



# **Oppaan tuottaminen**

## PHP Yii Framework

Jussi Hautala

Opinnäytetyö  
Toukokuu 2013  
Tietojenkäsittely  
Ohjelmistotuotanto

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittely  
Ohjelmistotuotanto

HAUTALA, JUSSI:  
Oppaan tuottaminen  
PHP Yii Framework

Opinnäytetyö 70 sivua, joista liitteitä 34 sivua  
Toukokuu 2013

---

Opinnäytetyön tavoitteena oli käydä läpi verkko-oppaan tuottamisessa huomioitavia asioita ja saada toteutettua oppaalle helppolukuinen ja selkeä rakenne. Työn aikana käsiteltiin PHP ohjelmistokehyksiä ja käytiin läpi niiden toimintaympäristön vaatimuksia. Vertailussa olivat mukana myös paikalliset WWW-palvelimet ja niiden valinnassa otettiin huomioon ohjelmistokehysten vaatimukset. Aiheen valintaan vaikutti oma kiinnostus ohjelmistokehyksiin.

Työn ohella tuotettiin verkko-opas Yii -ohjelmistokehyksen käyttämisestä WWW-ympäristössä. Oppaassa esitellään Yii -ohjelmistokehyksen toimintaperiaate ja kerrotaan hieman tarkemmin sen toimintatavasta. Asennusohje käy läpi asennuksen Windows -ympäristöön, jonka jälkeen koodiesimerkeillä on annettu esimerkkejä ohjelmistokehyksen toiminnasta. Työtä voivat käyttää alkeisoppaana ohjelmistokehyksistä kiinnostuneet henkilöt.

Ohjelmistokehyksen toimintaperiaatteista, asennuksesta ja käyttämisestä kertova opas on mahdollista jatkossa siirtää WWW-sivulle, jolloin se saadaan vielä tehokkaampaan käyttöön. Oppaaseen olisi mahdollista lisätä projekti, joka rakennettaisiin oppaan mukaan vaihe vaiheelta ja jossa pyrittäisiin käymään monipuolisesti ohjelmistokehyksen eri toimintoja ja toimintatapoja läpi.

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Business Information Systems  
Option of Software Engineering

HAUTALA, JUSSI:  
Producing a Guide  
PHP Yii Framework

Bachelor's thesis 70 pages, appendices 34 pages  
May 2013

---

The object of this thesis was to plan how to design a good web guide. This work contains basic information about web software frameworks and their operational environment requirements. In comparison were also associated web servers and framework requirements. The idea was born from the interest of producing websites.

During the work a web guide was produced on how to use the Yii framework and its working principle. Installation instructions are included for Windows environment and code examples help to learn the Yii framework. This guide can be used by people who are interested in web application frameworks.

In the future, this guide can be converted into a dynamic web-page. It ensures an even more effective use of the guide. The web page could contain the project, which was completed with the guide.

---

Key words: guide, framework, yii, php

## SISÄLLYS

1	JOHDANTO.....	5
2	OPPAAN TUOTTAMINEN VERKKOON .....	6
2.1	Julkaisuprosessi .....	6
2.2	Lähteet .....	6
2.3	Tekijänoikeudet .....	7
2.4	Julkaisumuoto .....	7
2.5	Verkkoteksti.....	9
2.6	Havainnollistaminen ja värit.....	10
2.7	Luettavuus.....	11
2.8	Tallennus.....	12
2.9	Kuvamateriaali.....	12
3	PHP WEB APPLICATION FRAMEWORK.....	13
3.1	Yii Framework.....	14
3.2	CakePHP.....	16
3.3	CodeIgniter .....	16
3.4	Zend .....	17
3.5	Symfony.....	18
4	WWW-PALVELIMET .....	19
4.1	WAMP.....	19
4.2	Wampserver.....	19
4.3	EasyPHP .....	20
4.4	XAMPP.....	21
5	TYÖYMPÄRISTÖ.....	22
5.1	Windows 7 .....	22
5.2	VirtualBox .....	22
5.3	Notepad++ .....	23
6	YII FRAMEWORK .....	24
6.1	MVC-Malli .....	25
6.2	TDD-Kehitys .....	27
7	TYÖVAIHEET .....	29
7.1	Valmistelut ja testikokoonpanon asennus.....	29
7.2	Yii -ohjelmistokehityksen asennus .....	30
7.3	Ympäristömuuttujat ja uusi projekti .....	30
8	POHDINTA.....	32
	LÄHTEET.....	34
	LIITTEET .....	36

## 1 JOHDANTO

WWW-sivujen tekemisessä on monia toiminnallisuuksia, joita käytetään yhä uudestaan ja uudestaan eri projekteissa. Aikaa kuluu todella paljon, jos nämä toiminnallisuudet tehdään joka kerta uudelleen. Toiminnallisuudet, jotka ovat uudelleenkäytettäviä, nopeuttavat työskentelyä huomattavasti ja tuovat tehokkuutta. Työkaluja on paljon tarjolla ja niitä löytyy mm. työpöytäsovelluksina hyvin paljon. Ne eivät kuitenkaan vielä auta tarpeeksi, että toiminnallisuuksia saataisiin tehokkaaseen käyttöön. Tässä haetaan nyt ohjelmistokehystä (Framework), jolla saadaan aikaan se haluttu tehokkuus ja ajansäästö.

Työn aikana vertaillaan eri ohjelmistokehyyksiä keskenään. Vertailujen jälkeen käydään hieman syvemmin läpi Yii -ohjelmistokehystä ja siihen liittyviä ominaisuuksia. Yii tarvitsee palvelimen toimiakseen ja työssä on vertailtu muutamaa paikallista palvelinohjelmistoa. Seuraava läpikäytävä asia on järjestelmä ja sen vaatimukset. Kun vaatimukset ja järjestelmään asennettavat ohjelmat on käyty läpi, kerron työn etenemisestä vaihe vaiheelta.

Työn aiheena on tuottaa opas ohjelmistokehyyksen käytöstä ja tutustua hieman sen toimintaperiaatteeseen. Ennen oppaan tekemistä on haettu tietoa siitä, kuinka on mahdollista saada aikaan selkeä ja luettava opas. Myös sisältöön ja ulkonäköön on pyritty vaikuttamaan, että lopputulos pysyisi selkeänä ja helposti luettavana. Samalla on pyritty huomioimaan lähteiden käyttöä ja tekijänoikeudellisia asioita. Työn aikana on otettu selvää erilaisista julkaisumuodoista. Toimeksiantajaa työlle ei ole, mutta se on toteutettu työnhakua varten. Opas on tämän raportin liitteenä ja se sisältää osittain samoja aiheita, mitä raportissa on esitelty (kts. Liite 1).

Lähteitä liittyen työhön löytyi paljon. Ohjelmistokehyykseen liittyen löytyi heti kaksi kirjaa, jotka käsittelevät aihetta alkeista hieman vaativampaankin käyttöön. Myös valmistajan sivuilla on hyvin materiaalia, minkä avulla pääsee alkuun. Kysymyksiä on mahdollista lähettää valmistajan foorumille, missä on aktiivinen yhteisö vastaamassa kysymyksiin. Aiheita hyvän oppaan tuottamiseen löytyi vähän. Suoraan itse aiheeseen ei ollut kirjoja, mutta aiheesta lähellä olevia kirjoja löytyikin jo muutama. Näistä valitsemalla halutut aiheet sai kasaan sopivan määrän aineistoa.

## 2 OPPAAN TUOTTAMINEN VERKKOON

Käyttäjätutkimuksen mukaan sisällöllä on suuri merkitys ja käyttäjät ovat siitä kiinnostuneita. Käyttäjän tullessa uudelle sivulle katse kääntyy suurimpaan sisältöalueeseen ja otsikoihin. Tämän jälkeen käyttäjä päättää onko sivu kiinnostava vai siirtyykö hän muualle. (Nielsen 2000, 100.) Aiempaan viitaten sisällöllä ja rakenteella on suuri merkitys verkko-oppaassa. Tässä luvussa käsitellään huomioon otettavia asioita oppaan suunnittelussa ja toteutuksessa.

### 2.1 Julkaisuprosessi

Verkkoon tulevan julkaisun vaiheet alkavat suunnittelusta, jatkuvat toteutukseen ja lopuksi julkaisuun. Suunnitteluun kuuluvat rakenteen ja sisällön suunnittelu. Tämän vaiheen aikana voidaan kirjoittaa synopsis siitä, millaista sisältöä julkaisu pitää sisällään, kenelle se tulee olemaan ja miksi julkaisu tehdään. Tämän perusteella voidaan arvioida työhön käytettävä aika. Rakennesuunnitelmalla voidaan hahmotella sivujen sisältö ja se on tavallisesti kaavio, joka sisältää kaikki sivut ja niiden mahdolliset yhteydet. Toteutusvaiheessa sivut koostetaan yhteen ja testataan. (Keränen, Lamberg & Penttinen 2006, 9.) Visuaalinen suunnittelu on hyvä tehdä kuvankäsittelyohjelmalla. Tähän suunnitelmaan tulevat mukaan käytettävät värit, typografia ja sommittelu. (Keränen ym. 2006, 10.)

Julkaisu tässä tapauksessa tarkoittaa julkaisun siirtämistä WWW-palvelimelle. Jos palvelin on vuokrattava, on hyvä selvittää millaiset rajoitukset on kyseisellä palvelimella käytössä. Tilantarve on hyvä tarkistaa, että tilaa on varmasti riittävästi julkaisulle. (Keränen ym. 2006, 14.)

### 2.2 Lähteet

Internetissä on valtavasti aineistoa, joka vaikeuttaa sieltä löytyvän aineiston luotettavuuden arviointia. Näistä julkaisuista puuttuvat myös usein julkaisupäivämäärät ja kirjoittajat. Asiantuntemus on täten vaikea arvioida. Vanhentunutta tietoa löytyy paljon ja

sivuja ei ole välttämättä päivitetty moneen vuoteen. Luotettavia sivuja yleisesti ovat yritysten ja organisaatioiden sivut, jotka usein ovat päivitettyjä ja näin ajan tasalla. Markkinointiteksteissä tieto ei ole objektiivista. Luotettavia tiedonlähteitä ovat sähköiset julkaisut, kuten tietopankit ja PDF-dokumentit. Näiden toteutuksessa on käytetty samaa periaatetta kuin painetuissa teksteissä. (Keränen 2007, 13–15.)

Muita lähteitä voi olla mm. kirjat ja lehdet. Kirjojen kanssa tulee pyrkiä siihen, että ne eivät olisi kauhean vanhoja ja tarkistaa, onko uusia painoksia saatavilla.

### **2.3 Tekijänoikeudet**

Tekijällä on omaan luovaan teokseensa tekijänoikeus. Teos voi olla kirjallinen, veistos, maalaus, piirros, valokuva yms. Tekijänoikeuslaki suojaa työn tekijän teoksia. Laki suojaa yhtäläillä myös ei ammatillisessa mielessä tehtyjä teoksia. Tekijänoikeus syntyy heti teoksen luomisvaiheessa tekijälleen. Jos tekijöitä on useampi, syntyy tekijänoikeus koko ryhmälle. Teosta koskevaan päätökseen tarvitaan kaikkia tekijänoikeuden omistajia. Tekijänoikeus kuuluu tekijälleen hänen koko elämänsä. Tekijänoikeuden haltijan kuoleman jälkeen alkaa suoja-aika, joka kestää pääsääntöisesti 70 vuotta. Toisen teosta ei saa kopioida ilman tekijän lupaa. Jos toisen teosta käytetään, on tekijällä oikeus saada siitä halutessaan korvaus. Tekijä voi myydä teoksensa käyttöoikeuden joko rajoitettuna, rajoittamattomana tai yksinoikeudella. Rajoittamattoman käyttöoikeuden kanssa ostaja voi käyttää teosta haluamallaan tavalla ja useita kertoja, mutta tekijällä on edelleen oikeus myydä teos toiselle osapuolelle. Rajoitetussa käyttöoikeudessa teoksen ostaja saa käyttää teosta vain tiettyihin käyttötarkoituksiin. Yksinoikeudessa ostajalla on ainoastaan oikeus käyttää teosta. (Keränen 2007, 151.)

### **2.4 Julkaisumuoto**

Toimisto-ohjelmilla on ohjelmistokohtaisia tiedostomuotoja dokumenttien tallentamiseen. Tämä vaikuttaa niiden avaamiseen muilla toimisto-ohjelmilla ja ne eivät välttämättä avaudu ollenkaan muilla kuin tiedoston alkuperäisellä ohjelmalla, jolla se on luotu. OpenOffice pystyy avaamaan Microsoft Officella tallennettuja dokumentteja, mutta tyyli ei välttämättä näy samalla tavalla, miten on tarkoitettu alkuperäisessä doku-

mentissa. Nykyään on yleistä käyttää toimisto-ohjelmien välillä XML-kieltä, jonka tarkoituksena on, että samoja tiedostoja voidaan avata ja käsitellä useilla ohjelmilla eri käyttöjärjestelmissä. Näihin lukeutuu nykyään myös verkkoselaimet. (Keränen 2007, 160.)

OpenDocument-tiedostomuoto (ODF, OASIS Open Document Format for Office Applications) on avoin tiedostomuotostandardi, joka kehitettiin alun perin OpenOfficea varten ja se perustuu XML-kieleen. Suljettuihin tiedostomuotoihin poiketen avointa tiedostomuotoa on eri ohjelmistovalmistajien mahdollista hyödyntää. Se soveltuu teksti-, taulukkolaskenta-, kaavio-, ja esitysgrafiikka-asiakirjoihin. OpenDocument on mahdollista muuttaa toiseen muotoon XML-kielen ansiosta. Nämä tiedostomuodot eivät kuitenkaan sovi julkaisuun, joten siihen on kehitetty omia tiedostomuotoja. (Keränen 2007, 160.)

Internet-jakeluun soveltuvia tiedostomuotoja ovat esimerkiksi Adoben PDF- ja Flash-tiedostomuodot. PDF (Portable Document Format) on ohjelmistoriippumaton siirrettävä tiedostomuoto. Se sopii sähköisten dokumenttien tallennus-, jakelu-, ja arkistointimuotoa. PDF on hyvin yleinen muoto internetin kautta jaettavissa materiaaleissa ja sitä käytetään yleensä paperimuotoisten julkaisujen sähköisinä versioina. Dokumentit voivat sisältää multimediaa, 3D-objekteja ja sähköisiä lomakkeita. Dokumentit on mahdollista suojata salasanalla ja näin rajoittaa niiden käyttöä ja muokattavuutta. Tulostamiseen saa myös asetettua rajoituksia. (Keränen 2007, 160.)

Dokumentti sisältää tekstissä käytetyn fontin, joten dokumentti aukeaa samanlaisena kaikissa ympäristöissä. Siitä johtuen PDF-dokumentti on parempi julkaisumuoto kuin toimisto-ohjelmien tiedostomuoto. PDF-dokumentteja voidaan luoda Adoben omaa ohjelmaa käyttäen tai virtuaalitulostimella, jolla saa esimerkiksi toimisto-ohjelmasta dokumentin suoraan PDF-muotoon. (Keränen 2007, 160–161.)

Useat sivut ovat tehty käyttäen HTML-kieltä. Se on yksinkertainen ja helposti opittava kieli, joka koostuu elementeistä ja attribuuteista, jotka kuvaavat sivun rakennetta. Selaimet eivät tarkista HTML:n oikeellisuutta, jotenka sivut voivat sisältää virheitä. Eri selaimissa sivut voivat näyttää ulkoasultaan erilaisilta, koska jokainen selain käsittelee ne omalla tavallaan. (Keränen ym. 2006, 30.)



XHTML on samankaltainen kieli kuin HTML ja XHTML-kieli sisältää samat elementit ja ominaisuudet. Ero tulee siitä, että XHTML ei hyväksy virheitä. Sitä käytetään XML-kielen säännöin, joten sivun tulee olla aina muotoiltu rakenteellisesti oikein. Tällä on pyritty siihen, että sivut näkyisivät myös muilla kuin tietokoneen selaimilla. (Keränen ym. 2006, 31.)

Omalle työlleni olen valinnut julkaisumuodoksi PDF-muotoisen dokumentin. Se on mahdollista avata useimmissa järjestelmissä ja on tarpeeksi kevyt. Myös päivitykset onnistuvat tarvittaessa. Tiedoston saa kopiosuojattua ja asetettua niin, että ainoastaan tulostaminen on mahdollista.

## 2.5 Verkkoteksti

Oppaassa peruselementtinä toimii teksti, jonka ensisijainen tarkoitus on viestinnällinen. Kirjoitustyylillä ja sisällöllä on merkitystä. Samasta aiheesta voi kirjoittaa monella tavalla riippuen siitä, miten halutaan vaikuttaa lukijaan. Teksti on graafinen elementti ja sillä on myös visuaalinen merkitys. Typografia tarkoittaa yhden kirjaimen ulkonäköä ja asemoinnilla kaikkien tekstien ja kuvien asettelua sivulla. Typografia tarkoittaa tekstin visuaalista ilmeen suunnittelua. Ulkoasuun vaikuttavat kirjasinkoko, kirjasinleikkaus ja kirjasintyyli. Tekstin lukeminen näytöltä tai paperilta eroaa monelta osin ja siinä onkin hyvä ottaa huomioon muutamia seikkoja. Näytöltä tekstin lukeminen sujuu hitaammin kuin paperilta luettaessa. Näytöltä luettaessa hidastavia tekijöitä ovat monitorin etäisyys, heijastukset ja tekstin vierittäminen näytöllä. Näytöltä luettaessa tekstiä myös muistaminen ja ymmärtäminen ovat vaikeampia kuin paperilta luettaessa. Tekstin lukemista näytöltä helpottavat seuraavat tekijät:

- Otsikoiden ja luetteloiden selkeä käyttö
- Otsikot auttavat dokumentin silmäilyssä ja kertovat, missä mikäkin aihe on
- Otsikko kertoo tiivistetysti sitä seuraavan sisällön.

Otsikoiden on tarkoitus myös herättää lukijan kiinnostus, koska niistä näkee, mitä aiheita käsitellään. Väliotsikoiden on myös tärkeä tulla esiin tekstistä yhtenäisellä tyylillään. Lyhyt ja ytimekäs teksti sopii näytölle hyvin. Suositeltava pituus yhdelle kappaleelle on alle 10 riviä, koska muuten kappaleista tulee liian pitkiä ja niitä on siten vaikea hahmottaa. Tekstin tulee olla selkeätä ja lauseet eivät saa olla liian pitkiä. Hyperteksti on digi-

taallinen teksti, joka sisältää linkkejä. Linkkien takana voi olla tekstikappaleita, dokumentteja tai muuta materiaalia. (Keränen 2007, 170–172.)

## 2.6 Havainnollistaminen ja värit

Havainnollistamiseen kuuluu viestin ymmärrettävyys, kiinnostavuus ja houkuttelevuus. Kuvien avulla voidaan konkretisoida abstraktia ja käsitteellistä. Tyhjää tilaa täytyy olla tarpeeksi kuvien ja elementtien välillä, se on oma elementtinsä, joka selkeyttää ja auttaa lukijaa pysymään mukana, eikä tule liian ahdasta muihin elementteihin verrattuna. (Lammi 2009, 56.)

Kuva jää tekstiä paremmin mieleen ja se voi toimia lukijalla muistivihjeenä. Kuva ei tuo tehostusta, jos se ei tue aihetta. Kuvan rajaaminen vaikuttaa tunnelmaan ja siihen, mitä kuvalla halutaan sanoa. Sivulla voidaan käyttää myös yksinkertaistettuja ikoneita kuvaamaan eri asioita ja kokonaisuuksia. Graafiset symbolit ovat yksinkertaistettuja grafiikkaobjekteja, joita voidaan käyttää havainnollistamiseen. (Lammi 2009, 57.)

Värit tehostavat viestintää. Niillä korostetaan, erotellaan, yhdistellään ja luokitellaan. Asioiden ryhmittely onnistuu väreillä. Esimerkiksi samanväriset objektit koetaan kuuluvan samaan ryhmään. Värien valinnat eivät ole yhdentekeviä, koska värit vaikuttavat voimakkaasti lukijaan. Sävyllä tarkoitetaan värin aallonpituutta, jota voidaan myös kutsua värin nimeksi. Kirkkaus on värin tummuusaste ja se kertoo, kuinka mustana tai valkoisena väri on. Kylläisyys on värin puhtautta ja väripitoisuutta. Väreillä voidaan ohjata katsetta korostettuihin elementteihin. On kuitenkin vältettävä liian voimakkaita värejä ja niiden yhdistelmiä. Tiettyä väriä tulee käyttää aina samaan asiaan, jotta yhdenmukaisuus säilyy sivujen välillä. On vaikea erottaa värejä, joiden kirkkausaste on sama. Tämä ilmiö tulee hyvin näkyviin, kun käytetään tummia ja vaaleita sävyjä. Värit koetaan erilaisiksi etäisyyksiltään. Lämpimät värit tuntuvat olevan lähempänä katsojaa kuin kylmät. Tästä johtuen lämmin väri tulisi sijoittaa etusijalle. Olennaisia asioita ei tulisi korostaa punaisen, sinisen ja vihreän eroilla, sillä kaikilla ei välttämättä ole hyvä värinäkö. (Lammi 2009, 66–73.)

## Värien merkitykset:

- keltainen: lämmin, myönteinen, eloisa
- oranssi: positiivinen, energinen, raikas
- punainen: aktiivinen, päällekkävyä, jännittävä
- ruskea: konservatiivinen, sosiaalinen, tunnollinen, turvallinen
- sininen: etäinen, rauhoittava, viileä
- vihreä: pirteä, rauhallinen, harmoninen
- violetti: arvokas, henkinen, mystiikka
- valkoinen: puhdas, ylellinen
- musta: ehdoton, synkkä, sci-fi, taiteellinen
- harmaa: itsenäinen, neutraali, yhdenmukainen.

## 2.7 Luettavuus

Kirjasintyytit luokitellaan niiden yhteisten piirteiden mukaan kirjasintyyliin. Perustietona olisi ainakin hyvä tietää antiikva- ja groteski-kirjaimet. Pääteellinen kirjasintyyli on antiikva (serif) ja Times New Roman on esimerkiksi sellainen. Pääteetön kirjasintyyli on groteski (sans serif), jossa kirjasinten viivat ovat suunnilleen yhtä paksuja. Tähän kuuluu esimerkiksi kirjasimet Arial, Tahoma ja Verdana. OpenType-fonttimuoto on Microsoftin ja Adoben kehittämä tekniikka, joka mahdollistaa fonttien toiminnan sekä Mac- että Windows-ympäristössä. Fontin käyttökelpoisuus riippuu sen luettavuudesta, millä tarkoitetaan sujuvuutta sekä eri kirjainmerkkien erottuvuutta toisistaan. Tähän voi antaa esimerkiksi ison i-kirjaimen, pienen l-kirjaimen ja numeron 1. Useita fontteja ei kannata käyttää saman työn sisällä, vaan korkeintaan kaksi fonttia riittää hyvin. Toinen fontti on tarkoitettu otsikoille ja alaotsikolle, toinen pidemmille tekstikokonaisuuksille. Hyvänä pidetään tapaa, jossa toinen fontti on antiikva ja toinen groteski-tyylinen fontti. Myös yhdellä fontilla saa näyttävän kokonaisuuden aikaiseksi. (Lammi 2009, 83–90.)

## 2.8 Tallennus

### **Etuja**

Tallentaessa jakelu PDF-muotoiseksi tiedostokoko pienenee verrattaessa toimisto ohjelmien tallennusmuotoihin. Fontit pysyvät samoina jokaisessa ympäristössä, koska ne on sisällytetty PDF:ään. PDF-tiedostoon ei jää ylimääräisiä kommentteja tai muistiinpanoja. (Lammi 2009, 239.) HTML-muotoisen julkaisun hyvinä puolina on sen päivitettävyys ja mahdollisuus lisätä dynaamista sisältöä.

### **Haittoja**

Tiedoston muokkaaminen ei ole mahdollista jälkikäteen. Jos virheen haluaa korjata, täytyy tehdä uusi versio. Tästä johtuen työstä voi syntyä useampia versioita. (Lammi 2009, 239.) HTML-kielen huonona puolena voidaan pitää sitä, että eri selaimet voivat näyttää sivun rakenteen hieman erilailla.

## 2.9 Kuvamateriaali

Kuva jää tekstiä paremmin mieleen ja se voi toimia lukijalla muistivihjeenä. Kuva ei tuo tehostusta, jos se ei tue aihetta. Kuvan rajaaminen vaikuttaa tunnelmaan ja siihen, mitä kuvalla halutaan sanoa. Graafiset symbolit ovat yksinkertaistettuja grafiikkaobjekteja, jotka voivat viitata johonkin abstraktiin käsitteeseen. Sivulla voidaan käyttää myös yksinkertaistettuja ikoneita kuvaamaan eri asioita ja kokonaisuuksia. (Lammi 2009, 148; 160.)

Valokuvan tai piirroskuvan käyttäminen riippuu tilanteesta. Opetusmateriaalissa voi yksinkertainen viivapiirros toimia valokuvaa paremmin. Käsien piirtämällä saadaan tuotua kuviin erilaista näkemystä. Skannattuja kuvia on mahdollista jatko käsitellä myöhemmin, jotta ne saadaan sopimaan paremmin sivulle. Internetissä on paljon erilaisia kuvapankkeja, joissa on paljon erilaista kuvamateriaalia. Toiset palvelut ovat ilmaisia ja toiset maksullisia. Kun käyttää kuvapankkien kuvia, täytyy muistaa tarkistaa käyttöoikeus huolella. (Lammi 2009, 153.)

### 3 PHP WEB APPLICATION FRAMEWORK

Web Application Framework (WAF) on WWW-sovelluskehys, joka on suunniteltu WWW-sovelluksien toteuttamiseen. Nykyään on tarjolla paljon avoimeen lähdekoodiin perustuvia sovelluskehyskä. Monet näistä käyttävät MVC-arkkitehtuuria. Ne mahdollistavat sovelluksen rakenteen standardisoinnin ja vähentävät kehitystyöhön käytettävää aikaa. (Okanovic 2011, 1315.)

Sovelluskehys sisältää useita komponentteja, joista syntyy uudelleenkäytettävä arkkitehtuuri. Se ohjeistaa luokkien sijoittelun ja määrittelee niiden vastuut ja yhteistyön. Kehittäjä räätälöi sovelluskehysten tietyksi sovellukseksi aliluokittamalla ja muodostamalla tapauksia sovelluskehysten luokista. (Okanovic 2011, 1315.)

Millainen on mielestäni hyvä ja tehokas ohjelmistokehys toimintaan, jossa WWW-sivuston sisältö on tärkeässä osassa? Pääpaino tulee pysymään sisällössä, eikä mene liikaa koodin ja toiminnallisuuksien määrittämiseen. Tarkoituksena on pääasiassa oppia ohjelmistokehysten toimintaa, ja olettaa, että aiempaa kokemusta asiasta ei ole. Kun periaate on hallussa, voi siirtyä käyttämään vaativampia toiminnallisuuksia. Ohjelmistokehysten tietoturvan on oltava ajan tasalla ja samoin myös dokumentointi. Mitä kattavampi dokumentointi on, sitä parempi. Valmiita toiminnallisuuksia, ohjelmakomponentteja, on hyvä olla mukana. Ne nopeuttavat työskentelyä jo pienempienkin toiminnallisuuksien kanssa.

Ohjelmistokehyskä on vaikea vertailla. Kaikilla ovat omat näkemyksensä siitä, mikä on hyvää ja mikä ei. Joku saattaa tarvita toista ominaisuutta, kun toinen ei tarvitse sitä ollenkaan. Toisille nopeus ja tehokkuus ovat avainasemassa, kun joku muu arvostaa vaativampia ominaisuuksia. Vertailu on täten vaikeaa, kun kaikkien kriteerit eivät kohtaa. Mikä vertailumenetelmä sopii vertailuun, siitä on jokaisella oma näkemys. Toki voidaan vertailla tiedostojen kokoa, nopeutta ja muuta vastaavaa, mutta se ei kerro läheskään kaikkea oleellista. Näihin voidaan luetella mm. tietoturva, käytännöllisyys ja arkkitehtuurin selkeys.

Ohjelmistokehysten testaamiseen voidaan käyttää samoja ohjelmia, joita käytetään WWW-palvelimien testaamiseen. Näillä testeillä saadaan viitteitä eri ohjelmistokehys-

sien nopeudesta ja tehokkuudesta. Keinoja, joilla voidaan testata ohjelmistokehystä ovat mm. sivulatausten määrä, palvelimen keskimääräinen raskaus, muistin käyttö, vasteajat, funktioiden kutsujen määrä ja tiedostojen yhteismäärä. Testauksen aikana testikokoonpanon pitää olla jokaisen testattavan kohdalla täsmälleen sama. Huomioon tulee ottaa myös verkkokortti ja sen nopeus, eikä sitäkään saa vaihtaa kesken testauksen. Testissä voidaan käyttää joko ihan tavallista ”*Hello World*” -sivua, jossa on ainoastaan kyseinen teksti. Siinä tapauksessa testi ei kerro mitään tietokannan nopeudesta. Testin pitäisi olla mahdollisimman monipuolinen. Tai sitten voidaan testata jokaista ominaisuutta erikseen ja tehdä niistä yksi raportti. Menetelmä on hidas, jos käydään jokaisesta ohjelmistokehystä yksi kohta kerrallaan ja jos käytössä on vain yksi palvelin.

Työhön on valittu viisi eri PHP ohjelmistokehystä. Valinta on tehty viidestä eniten ääniä saaneesta ohjelmistokehystä (PHP Frameworks 2013). Valintaan vaikutti niiden suosio ja se, että ne löytyvät useista eri ohjelmistokehysten vertailuista.

### 3.1 Yii Framework

Yii Framework-projekti aloitettiin tammikuussa vuonna 2008 Qiang Xuen toimesta. Alun alkaenkin se on ollut avoimeen lähdekoodiin perustuva ohjelmistokehysprojekti. Ensimmäinen alkeisversio tuli lokakuussa 2008 ja versio 1.0 joulukuussa jo samana vuonna. Nimi Yii tulee sanoista *'yes, it is'* ja se tarkoittaa helppoa, tehokasta ja laajennettavaa. Yii on tehokas komponentti-pohjainen WWW-ohjelmistokehys, joka on kirjoitettu PHP5-kielellä. Yii helpottaa suurten WWW-ohjelmien ja sovellusten luontia ja ylläpitoa, ja siihen on helppo liittää valmiita tai itse tehtyjä liitännäisiä. (Winesett 2010, 7–8.)

Yiitä on kehitetty yhdeksi nopeimmista ohjelmistokehysistä. Nopeudella tässä tapauksessa tarkoitetaan sivulatauksia. Uusia toiminnallisuuksia saadaan helposti automaattisella koodin generoinnilla ja se sisältää jo valmiiksi paljon ominaisuuksia. Hyvin tarjolla aiheeseen liittyviä kirjoja, mutta eivät ole nykyisten Yiin versioiden tasolla. Käyttäjän todennus ja navigaatiopalkki löytyvät valmiina mallipohjassa, johon on helppo tutustua.



about downloads

## The **Fast**, **Secure** and **Professional** PHP Framework +1 2.6k

Yii is a **high-performance** PHP framework best for developing Web 2.0 applications.

Yii comes with **rich features**: MVC, DAO/ActiveRecord, I18N/L10N, caching, authentication and role-based access control, scaffolding, testing, etc. It can reduce your development time significantly.

KUVA 1. Yiiframework (Yii Software LLC 2013b)

### 3.2 CakePHP

Projekti on aloitettu huhtikuussa vuonna 2005 Michal Tatarynowiczin toimesta ja versio 1.0 julkaistiin toukokuussa 2006. CakePHP käyttää arkkitehtuurina MVC-mallia. Koodin generoinnilla luodaan nopeasti prototyyppisiä, jotka keskenään tulevat muodostamaan valmiin kokonaisuuden. Asetuksissa ei käytetä XML- tai YAML-tiedostoja. CakePHP toimii MIT-lisenssin alla, joka sopii kaupallistenkin projektien tekemiseen. Sisältää valmiina tietoturvakomponentteja, jotka ovat valmiita käytettäväksi. (Cake Software Foundation 2013.)

CakePHP lukeutuu hitaimpiin vertailussa mukana oleviin ohjelmistokehyksiin. Se on vaikea oppia, jos taustalla ei löydy valmiiksi osaamista jostain ohjelmistokehyksestä. Ohjelmistokehyksen käyttö on hieman kömpelöä. Lomakkeiden tarkastusperiaate on hieman oudolla tavalla toteutettu. Taustalla toimii suuri yhteisö ja saatavilla on paljon valmiita lisäpalikoita laajentamaan ohjelmakehystä. Tarjolla on runsaasti ladattavia koodiesimerkkejä.



KUVA 2. CakePHP (Cake Software Foundation 2013)

### 3.3 CodeIgniter

Projektiä varten on perustettu EllisLab vuonna 2002, jonka jälkeen aloittivat muutaman vuoden opiskelun ohjelmistokehyksen suunnittelusta ja toteutuksesta ja lopulta syntyi opituilla tiedoilla ja taidoilla CodeIgniter-ohjelmistokehys. CodeIgniter on nopea, sisältää hyvän dokumentoinnin ja on helppo oppia. Suunnittelussa on pyritty pitämään ohjelmistokehyksen tilantarve mahdollisimman pienenä. Taustalla toimii suuri yhteisö,



jotenka myös valmiita lisätoimintoja on saatavilla heidän toimestaan. CodeIgniter sisältää vähemmän ominaisuuksia kuin Yii. Jos haluaa kaiken hyödyn irti, täytyy ladata erillinen lisäosa, joka on maksullinen. (Ellis Lab Inc 2013a.)



KUVA 3. Codeigniter (Ellis Lab Inc 2013b)

### 3.4 Zend

Zend-projekti on aloitettu vuoden 2009 elokuussa ja ensimmäinen julkaisu Zend-ohjelmistokehys 2.0:sta tuli elokuussa vuonna 2010. Zend on avoimeen lähdekoodiin perustuva ohjelmistokehys, joka käyttää PHP 5.3-versiota toiminnoissaan. Arkkitehtuurina käytössä on MVC-malli. Zend on tarkoitettu käytettäväksi laajoille sivustoille, jossa on paljon erilaisia toiminnallisuksia. Maksullisella lisäosalla saa käyttöön osia, jotka laajentavat käyttömahdollisuuksia ja antavat huomattavan määrän lisämahdollisuuksia sivujen ja sovellusten tekemiseen. (Zend Technologies Ltd 2013b.)



KUVA 4. Zend framework (Zend Technologies Ltd 2013a)

### 3.5 Symfony

Kehitys aloitettiin vuonna 2005 ja kirjoitushetkellä viimeisin vakaa versio on vuodelta 2012 versionumerolla 1.4. Marraskuusta 2012 ensimmäinen versio Symfonystä ei saanut enää päivityksiä ja siirryttiin uuteen projektiin nimeltä Symphony2. Ohjelmistokehitys toimii GPL-lisenssin alla ja perustuu avoimeen lähdekoodiin. (Fabien Potencier 2013a.)



KUVA 5. Symfony (Fabien Potencier 2013b)

## 4 WWW-PALVELIMET

### 4.1 WAMP

WAMP on nimitys ohjelmistokokonaisuudelle, joka sisältää kaikki tarvittavat WWW-sovellukset, joiden avulla pystytään paikallisesti pyörittämään WWW -palvelinta. WAMP on muunnos LAMPista, joka on vastaava ohjelmistokokonaisuus Linux-ympäristöön. WAMP tulee sanoista Windows Apache MySQL PHP, Perl ja/tai Python. WAMP:iin pohjautuvan Wampserverin mukana tulee phpMyAdmin-työkalu, jonka avulla onnistuu esimerkiksi MySQL-tietokannan luominen, muokkaus ja tarkastelu paikallisesti. Mukana on työkaluja, joiden avulla asetusten määrittäminen on helppoa, eikä välttämättä tarvitse mennä koskemaan erikseen asetustiedostoihin. Tämän palvelimen avulla pystyy paikallisesti pyörittämään mm. Apachea, PHP:ta ja MySQL:ää. (Aviointi 2011.)

### 4.2 Wampserver

Wampserver on Windows-järjestelmälle kehitetty WWW-palvelinympäristö. Sen avulla voidaan luoda WWW-sovelluksia. Wampserverin paketissa mukana tulevat Apache2, PHP, ja MySQL. Wampserver on pyritty rakentamaan helpoksi käyttää. Työskentelyä helpottamaan on tehty valikko, jonka kautta onnistuu mm. palvelimen offline/online-tilan vaihtaminen ja palvelimen asetuksille on oma valikko. Valikon kautta on myös mahdollisuus asentaa Apache, MySQL ja PHP-jakeluita ja luoda aliaksia. (Alter Way 2013.)



KUVA 6. Wampserver (Alter Way 2013)

### 4.3 EasyPHP

EasyPHP on WAMP-paketti, joka sisältää palvelinpuolen komentosarjakielenä PHP:n, WWW-palvelimena on käytössä Apache ja tietokantana MySQL. Mukana on kehitystyökalu phpMyAdmin, jonka avulla voidaan hallita tietokantoja. Virheiden etsintään on Xdebug-työkalu. EasyPHP:hen on mahdollista asentaa moduuleita, joiden avulla saadaan monipuolistettua käytössä olevien työkalujen määrää. Asennuksen voi hoitaa myös kannettavana versiona. Tällä tavoin ohjelmaa voidaan käyttää esimerkiksi USB-muistitikulta. Ohjelma on tarkoitettu ainoastaan kehitystyöhön, ei tuotantokäyttöön tai valmiiden sivujen isännöintiin. (Laurent, Emmanuel, Thierry 2013b.)



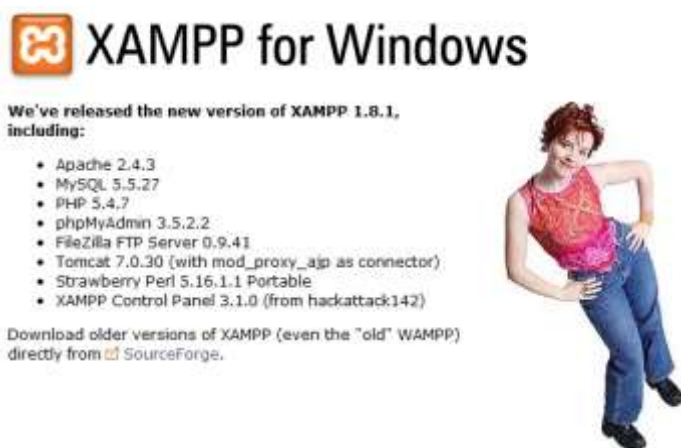
KUVA 7. EasyPHP (Laurent A. EasyPHP 2013a)

## 4.4 XAMPP

XAMPP on avoimeen lähdekoodiin perustuva paikallinen WWW-palvelin, joka sisältää mm seuraavat paketit (Kai 2013):

- Apache
- MySQL
- PHP
- PEAR
- PERL
- OpenSSL
- FileZilla
- FTP Server

XAMPP tulee sanoista X (*cross-platform*) Apache MySQL PHP Perl. XAMPP-jakelun saa Windows-, Linux-, Mac OS X- ja Solaris-järjestelmiin. Windows-jakelu on tuettu Windows 2000, 2003, XP, Vista ja 7-versioissa. XAMPP:in ideana on alun perin ollut rakentaa helppo jakelu aloittaville kehittäjille, minkä kautta kehittäjien on ollut helppo tutustua Apachen maailmaan. XAMPP toimii GNU-lisenssin alla, ja on näin ilmainen lisenssin ehtojen mukaisesti. (Softonic International S.L. 2013.)



KUVA 8. XAMPP (Softonic International S.L. 2013)

## 5 TYÖYMPÄRISTÖ

### 5.1 Windows 7

Windows 7 -käyttöjärjestelmä julkaistiin kuluttajille 22. lokakuuta 2009 (Sanoma News Oy 2009). Tässä projektissa on käytössä versio 7, vaikka Windows 8 olisi ollut myös vaihtoehtona. Tosin Windows 8 ei olisi tuonut mitään lisää omaan projektiin ja Windows 7:sta löytyy kaikki tarvittava ja siihen saa uusimmat päivitykset valmistajalta. Toisena vaihtoehtona käyttöjärjestelmälle oli Ubuntu, jonka saa myös hyvin helposti asennettua VirtualBoxin kautta käytettäväksi.

### 5.2 VirtualBox

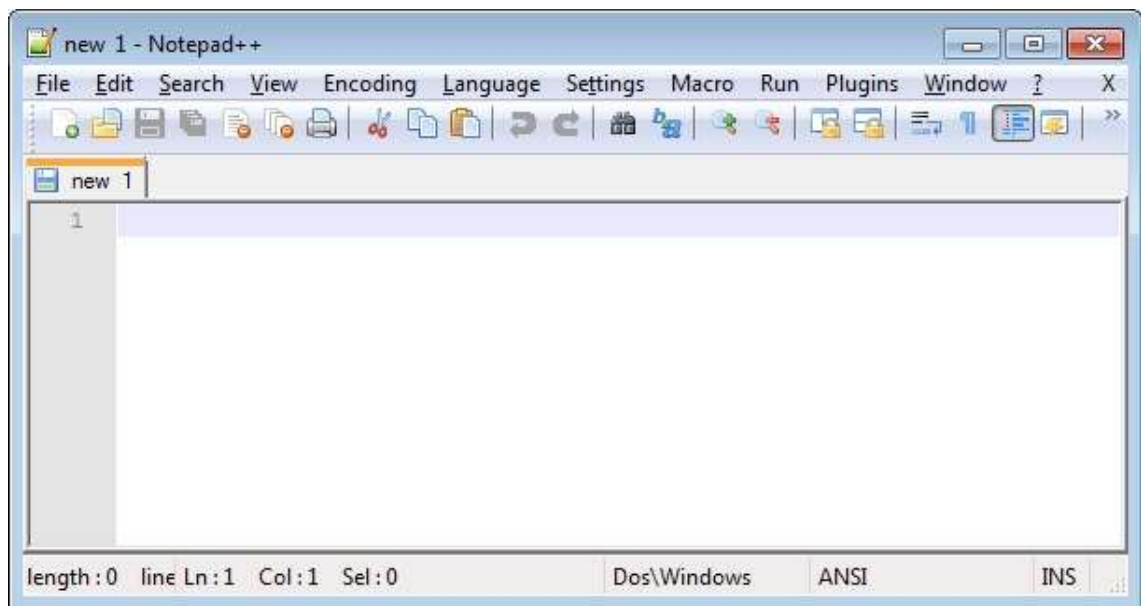
Ohjelma sai alkunsa saksalaisen Innotek GmbH-yrityksen toimesta ja helmikuussa 2008 Sun Microsystems osti yrityksen itselleen (The Register 2008). VirtualBox on virtuaalisointiin tarkoitettu ohjelma x86-, AMD64- ja Intel64-järjestelmiin. Se on tarkoitettu niin yritys kuin yksityiseenkin käyttöön. Ohjelma toimii GPL-lisenssin alaisuudessa. Ohjelmaa voidaan käyttää palvelinkäytössä, työpöytäkäytössä ja upotetuissa järjestelmissä. Tuettuja käyttöjärjestelmiä ovat mm. Windows, Linux, Macintosh, Solaris ja OpenBSD. Ohjelmaa kehitetään koko ajan lisää ja siihen tehdään lisäosia jatkuvasti yhä enemmän. Myös tuettuja järjestelmiä pyritään lisäämään jatkuvasti. (Oracle Corporation 2013.)

Käytin työssäni VirtualBoxia, joka toimii Windows-käyttöjärjestelmän päällä. VirtualBoxiin olin asentanut puhtaan asennuksen Windows 7 -käyttöjärjestelmästä. Tämä sen vuoksi, että saadaan puhdas asennus, joka takaa sen, että häiriötekijöitä on mahdollisimman vähän. Näin saadaan asennettua kaikki ohjelmat alusta alkaen ja saadaan tietoon mahdolliset lisäosat ja korjauspaketit, jotka täytyy asentaa työympäristön toiminnan kannalta. Kirjoittaessa uusin versio ohjelmasta oli 4.2.10.

### 5.3 Notepad++

Notepad++ on avoimeen lähdekoodiin perustuva tekstieditori, joka tukee useita eri ohjelmointikieliä. Ohjelma on kirjoitettu C++ -ohjelmointikieltä käyttäen. (Don Hon 2013.)

Ohjelma osaa muotoilla koodin helpottamaan lukemista värien avulla. Näitä kutsutaan teemoiksi ja niitä on mahdollista mukauttaa myös itse. Ohjelma on erittäin pieni ja kevyt, joten se soveltuu hyvin käytettäväksi tässä projektissa. Toinen vaihtoehto olisi ollut käyttää jotain kehitysympäristöä, joka olisi samalla tuonut lisää työkaluja helpottamaan työskentelyä. Yksinkertaisuus oli kuitenkin tässä vaiheessa itselle tärkeässä osassa. Samalla säästyi kehitysympäristön asennukselta, mahdollisilta lisäosien asennukselta ja pystyi keskittymään itse asiaan. Kirjoittaessa uusin versio ohjelmasta oli 6.3.



KUVA 9. Notepad++ (Don Hon 2013)

## 6 YII FRAMEWORK

### **Yii is easy**

Yiin toimintaan tarvitaan sen ohjelmakehyksen ydintiedostot ja WWW-palvelin, joka tukee versiota PHP 5.1.0 tai uudempaa. Yiin kanssa työskentelyyn on hyvä olla osaamista PHP:sta ja olio-ohjelmoinnista. Yii-sovelluksia rakennettaessa pääosassa ovat PHP-luokkien ylläpitäminen ja uusien luokkien luominen. (Winesett 2010, 8.)

### **Yii is efficient**

Yii on korkean suorituskyvyn komponenttiperusteinen ohjelmistokehys WWW-sovellusten kehittämiseen jokaisessa mittakaavassa. Kehitysprosessi nopeutuu koodin uudelleenkäytöllä. Yii on suunniteltu auttamaan DRY (Don't Repeat Yourself) kehityksessä. Kaikki Yii -sovellukset on rakennettu käyttäen MVC-arkkitehtuuria. Tämä estää toistoa ja auttaa edistämään koodin uudelleenkäyttöä ja helpottaa ylläpitoa. Mitä vähemmän koodia tarvitsee kirjoittaa, sitä helpompaa on ylläpito ja päivitys. Ohjelmistokehys on myös tehokas käyttää ja sen suorituskyky on optimoitu, jotta siitä saataisiin yksi tehokkaimmista PHP ohjelmistokehyksistä. (Winesett 2010, 8–9.)

### **Yii is extensible**

Suunnittelu on hoidettu huolellisesti, että lähes jokainen pala Yiin koodia voidaan laajentaa ja räätälöidä lähes mihin tahansa tarpeeseen tai vaatimukseen (Winesett 2010, 9).

Yiin huonona puolena voidaan pitää sen eri versioiden yhteensopimattomuutta. Jos on aloittamassa opiskelemaan ohjelmistokehystä kokonaan uutena asiana ja ottaa kirjan opintojen avuksi, kannattaa alusta asti käyttää samaa versiota, jota kirja tai mahdollinen lähde käyttää. Näin voidaan varmistua, että yhteensopivuusongelmia ei tule.



## 6.1 MVC-Malli

MVC-mallin avulla saadaan eroteltua käyttöliittymä sovellusalueiedoista. MVC-malli jakautuu kolmeen osaan, jotka ovat **malli**, **näkymä** ja **käsittelijä**.

**Mallin** (Model) tehtävänä on ylläpitää tilaa ja määrittää tiedon väliset suhteet tietokantatauluissa. Se hoitaa myös kirjoittamisen ja lukemisen tietokantaan. Jokaisella tietokantataululla on oma mallinsa ja ne ovat nimetty tietokantataulun mukaan. Mallin toimintaan saamiseksi ei tarvitse välttämättä kirjoittaa yhtään koodia. Sen saa generoitua valmiina käyttäen komentorivityökalua yiic, tai selaimessa gii-moduulia. Mallilla on kaksi tärkeätä tehtävää, jotka ovat tiedon haku ja tallennus. Tietokantahakuun ei tarvitse kirjoittaa erillisiä kyselyjä, vaan ne onnistuvat hyvin yksinkertaisella haulla. Jos taulusta halutaan hakea kaikki tieto, käy se lauseella `ModelName.FindAll`. (Winesett 2010, 10.)

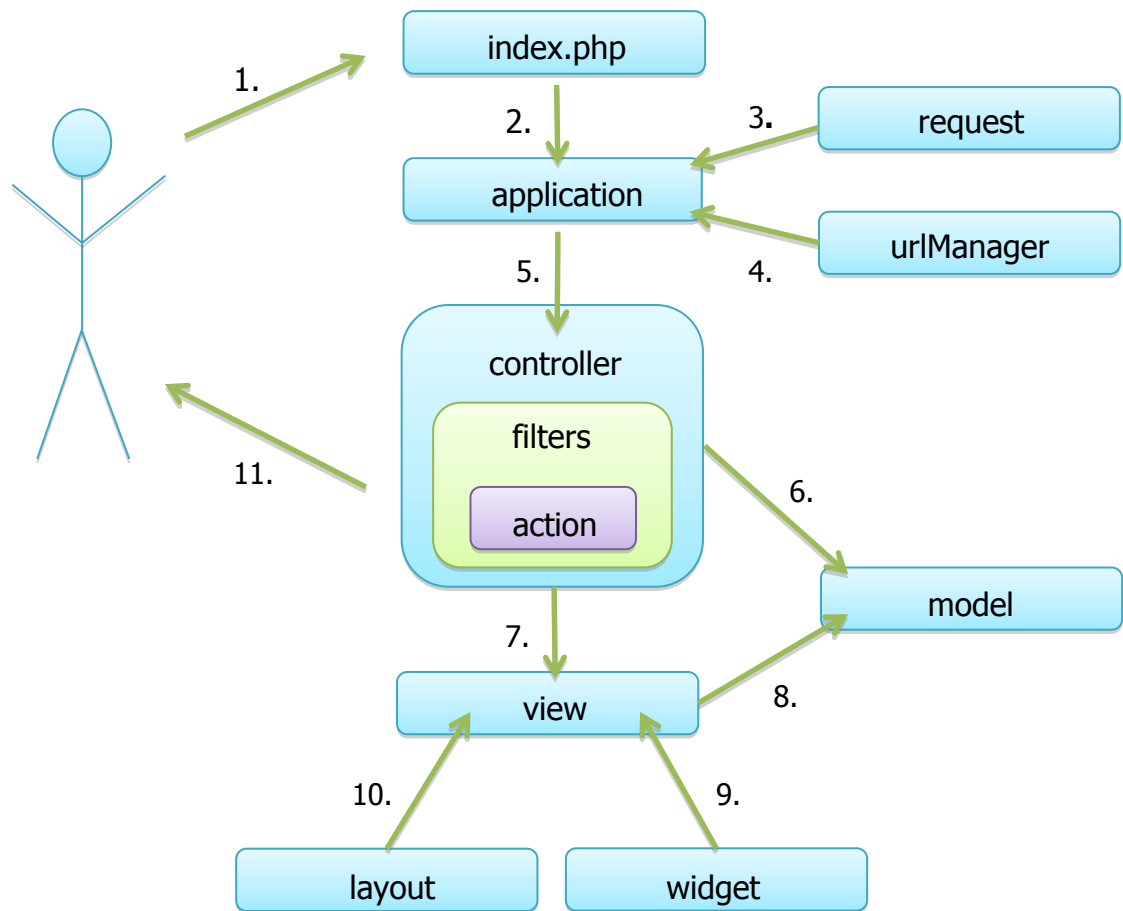
**Näkymä** (View) vastaa tyypillisesti käyttöliittymän muodostamisesta, joka perustuu tietoon mallissa. Näkymä on PHP-komentosarja, joka sisältää käyttöliittymään liittyviä elementtejä, jotka ovat yleensä rakennettu HTML:ää käyttäen, mutta ne voivat sisältää myös PHP-lauseita. Nämä PHP-lauseet ovat yleensä hyvin yksinkertaisia ehtoja tai silmukoita, jotka viittaavat muihin Yiiin elementteihin, kuten HTML-apuluokkaan tai esikäännettyihin ohjelmakomponentteihin. Näkymä saa tietoja käsittelijältä ja näyttää ne käyttäjälle. Tämä osio on ainoa, jossa käytetään myös HTML-, CSS- ja JavaScript- ym. kieliä. (Winesett 2010, 10.)

**Käsittelijä** (Controller) ohjaa reititetyt pyynnöt ja on vastuussa käyttäjän syötteistä, vuorovaikutuksessa mallin kanssa ja päivittää näkymää tarpeen mukaan. Käsittelijä on esiintymä `CController`-luokasta tai sen lapsiluokasta. Kun käsittelijä saa toiminnan, suorittaa se pyydetty tehtävät, jotka ovat vuorovaikutuksessa tarvittaviin malleihin, jonka jälkeen se muodostaa näkymän. Käsittelijää kutsutaan aina kun annetaan osoiteriville uusi osoite. Se voi lukea tiedot POST-, GET- ja SESSION-muuttujista ja käsitellä tietokantoja. MVC-arkkitehtuurin kanssa osoitteilla on tietty rakenne. Ne sisältävät parametrin, joka kertoo, mitä käsittelijää tullaan käyttämään. Kaikki navigointi hoidetaan parametrin ja käsittelijän kautta. (Winesett 2010, 10.)

Kuvassa 10 on esimerkki Yii:n tavasta käsitellä käyttäjän pyyntöä:

1. Käyttäjä avaa sivun <http://www.example.com/index.php?r=post/show&id=1> ja WWW-palvelin käsittelee *bootstrap*-komentosarjan, joka sijaitsee `index.php` -tiedostossa projektin juurihakemistossa.
2. *Bootstrap*-komentosarja luo `Application`-luokan istunnon ja suorittaa sen.
3. `Application`-luokan istunto pitää hallussaan yksityiskohtaista tietoa käyttäjän pyytämästä tiedosta, jonka se saa `Application`-komponentilta (*request*).
4. `Application`-luokan istunto määrittää `Controller`in ja *actionin* pyynnöt sovelluksen komponentilla `urlManager`.
5. `Application` luo istunnon pyydetyistä ohjaimista myöhempää käyttäjän pyyntöä varten. `Controller` määrittää toiminnon *show*, joka viittaa metodiin `actionShow`, joka sijaitsee `Controller`-luokassa. Tämän jälkeen `Controller` luo ja suorittaa filterit (esim. pääsyn hallinta, benchmarkkaus), jotka ovat yhdistettynä tähän toimintoon.
6. Toiminto lukee `Post`-mallin tietokannasta, jonka ID on 1.
7. Toiminto muodostaa näkymän, jonka nimi on *show*, joka on `Post`-mallin mukana.
8. Näkymä lukee ja näyttää `Post`-mallin attribuutit.
9. Näkymä suorittaa tarvittavat ohjelmakomponentit.
10. Näkymän muodostamisen tulos on upotettu sivupohjaan.
11. Toiminto suorittaa loppuun näkymän muodostamisen ja näyttää tuloksen käyttäjälle.

Kohdassa 1. mainittiin *bootstrap*-komentosarja. Se on pieni ohjelma, joka lataa tilanteeseen sopivan ohjelman. Se on hieman samanlainen kuin BIOS. Sen tehtävänä on käsitellä kaikki dynaamiset pyynnöt, jotka tulevat palvelimelle ja suorittaa MVC-mallin, jotta jatkossa yhden komponentin tai sovelluksen toiminnallisuutta voidaan muuttaa muuttamatta koko ohjelman koodia. (Jamsi 2013.)



KUVIO 1. MVC-malli Yiissä (Yii Software LLC 2013c)

## 6.2 TDD-Kehitys

Yii sisältää valmiiksi integroituna tuen TDD (Test Driven Development) -testaukseen. Pienissä projekteissa testaus ei ole välttämätöntä, mutta projektin suurentuessa testauksen tärkeys nousee esiin. Mitä enemmän eri toiminnallisuuksia ja niiden välisiä toimintoja tulee olemaan, sitä enemmän yhden toiminnallisuuden vajaa toiminta vaikuttaa koko muuhun ohjelmaan. (Makarov 2011.)

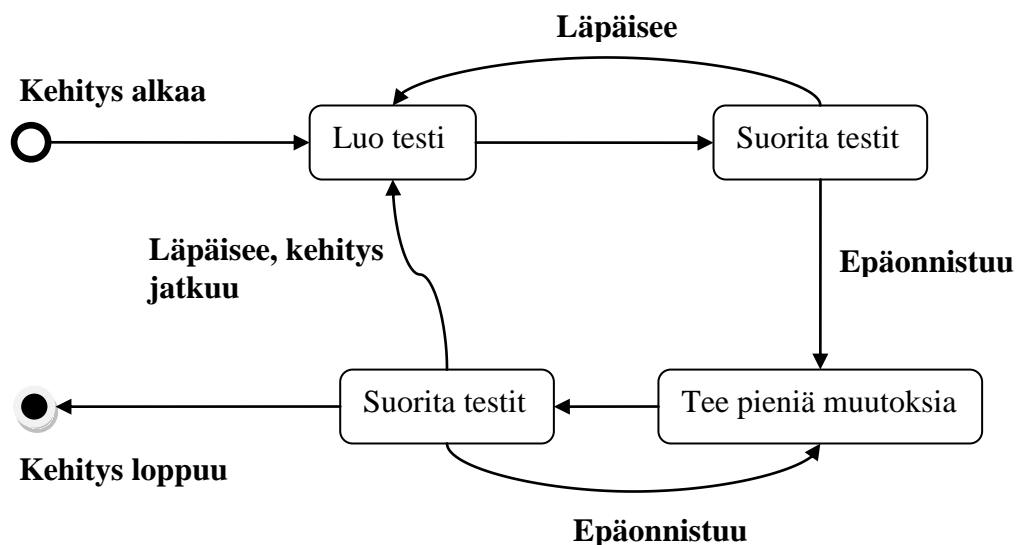
TDD-kehitys on ohjelmiston kehitysmenetelmä. Sen pääperiaate on, että kirjoitetaan nopea testi, joka ei mene läpi. Testi suoritetaan ja huomataan, että se ei mene läpi. Lisätään tarvittava määrä koodia, jotta testi saadaan menemään läpi. Suoritetaan testi ja katsotaan, että se menee läpi. Karsitaan koodista kaikki ylimääräinen pois ja ajetaan testi

vielä niin, että se menee läpi. Nämä kohdat toistuvat koko kehitysprosessin ajan. Tarkoituksena on kirjoittaa mahdollisimman pieniä kokonaisuuksia ja testata jatkuvasti, jotta virheitä ei pääsisi ohjelmaan missään välissä. (Winesett 2010, 43.)

Tärkeää on, että ensin kirjoitetaan testi ja vasta sen jälkeen varsinainen toiminto. Ohjelma luodaan testien avulla ja tarkoitus on mennä koko prosessi läpi suunnitelman mukaan. Testauksen kanssa vietetään paljon aikaa. Yhden virheen korjaaminen voi johtaa toiseen virheeseen. Virheitä voi tulla vastaan enemmän, mitä aluksi oli raportoitu. Aina ei voi varmaksi sanoa, löytyykö ohjelmasta ongelma, joka johtaa koko ohjelman kaatumiseen. Monet kokeneet testaajat sanovat, että lyhyet jaksot johtavat vähempiin ongelmiin. Aggressiivinen testaus säästää aikaa ja ponnisteluja. Ikävää ja virhealtista työtä ei tule jatkaa tai toistaa.

TDD-kehitys on tapa rakentaa ohjelmistoa vaiheittain. Vaikka ohjelmakoodia ei olisi kirjoitettu yhtään, on ensimmäinen testi tässä vaiheessa jo tehty. Ensimmäisen testin on tarkoitus aina epäonnistua. Testit ovat pieniä ja ne ovat automatisoitu. Testauksen automatisointi on TDD-kehityksen idea. Aina kun tuotteen koodi laajenee, muuttuu samalla yksikkötestauksen sarja. Aina kun koodiin tulee muutoksia, suoritetaan myös testit, jolla testataan uuden koodin toimivuus ja yhteensopivuus vanhan koodin kanssa

TDD-kehitys ei ole testaustekniikka, vaan se pyrkii luomaan useita arvokkaita automatisoituja testejä. Se on enemmän tehty ratkaisemaan ohjelman ongelmia. Se auttaa tekemään tärkeitä päätöksiä. Testit antavat selkeitä virheitä, kun joku asia ei toimi oikealla tavalla. (James 2011.)



KUVIO 2. TDD-kehityksen vaiheet (James 2011)

## 7 TYÖVAIHEET

### 7.1 Valmistelut ja testikokoonpanon asennus

Tässä osiossa käydään läpi Yiin asennus Windowsille, ja sama osio löytyy myös op-  
paasta, joka on kirjoitettu ohjemaiseksi ja siinä käydään syvällisemmin kohdat vaihe  
vaiheelta. Käyttöympäristön asennus alkaa ohjelmasta VirtualBox, johon asennetaan  
puhdas asennus käyttöjärjestelmästä. Käyttöjärjestelmänä toimii Windows 7, johon ei  
ole asennettu mitään ohjelmia tai sovelluksia. Järjestelmästä löytyy ainoastaan mukana  
valmiiksi tulleet ohjelmat ja sovellukset. Ydintiedostot löytyvät Yiin sivuilta, ja koska  
käytössä on vanhempi versio ohjelmistokehyksestä, pitää tiedostot hakea hieman kau-  
empaa arkistosta. Sivulla on listattu eri versiot ohjelmistokehyksestä ja listan alta löytyy  
linkki arkistoon, mistä löytyy vanhat versiot. Asennuspaketti on nimetty  
*1.1.12.b600af.zip*, joka sisältää Yiin version 1.1.2 ydintiedostot (Yii Software LLC  
2013a).

Kun on ladattu ydintiedostot, tarvitaan seuraavaksi WWW-palvelin, joka toimii paikal-  
lisesti. Tähän on valikoitunut Wampserver, joka sisältää kaikki tarvittavat ohjelmat Yii -  
ohjelmistokehyksen toimintaan. Ennen palvelimen asennusta täytyy asentaa Visual C++  
2010 SP1 -paketti, jotta Wampserver saadaan toimimaan. Wampserver voi vaatia toimi-  
akseen näitä lisäkirjastoja, joita ei tule Windowsin asennuksen yhteydessä. Tämän jäl-  
keen asennetaan ensin Visual C++ -paketti, jonka jälkeen voidaan asentaa Wampserver.  
Asennuksen aikana ohjelma kysyy oletusselainta ja jos sitä ei aseteta, käytetään oletuk-  
sena Internet Explorer -selainta. Tässä vaiheessa voidaan valita käyttöön jokin muu se-  
lain. Selaimeksi valikoitui Google Chrome. Seuraavana asennus kysyy verkon käyttöoi-  
keuksia. Jos ei ole varma, niin tämän kohdan voi jättää huomioimatta ja jatkaa eteen-  
päin. Asennuksen jälkeen Wampserver käynnistyy ja on heti käyttövalmis. Palvelimen  
kuvake pitäisi näkyä vihreänä. Jos näin ei ole ja kuvake on keltainen, on palvelimen  
toiminnassa jotain vikaa. Uudelleenasennus saattaa korjata ongelman. Tekstieditoriksi  
on valikoitunut Notepad++ sen helpposta, karsitusta ja kevyestä rakenteestaan johtuen.  
Sillä onnistuu kaikki tarvittava, mitä projektin aikana tarvitsee tekstin muodossa muoka-  
ta.

Testikokoonpano oli tässä vaiheessa seuraava:

- 64bit Windows 7 (VirtualBoxin kautta)
- Visual C++ 2010 SP1
- Wampserver (64bit & Apache 2.4) 2.2E
- Yii Framework 1.1.2
- Notepad++ 6.2.1
- Google Chrome 23.

## 7.2 Yii -ohjelmistokehyksen asennus

Ohjelmistokehyksen asennus onnistuu helposti. Ohjelmistokehyksen paketti vain puretaan palvelimen WWW-hakemistoon, jonka jälkeen on mahdollista tarkistaa järjestelmän yhteensopivuus ohjelmistokehyksen kanssa. Valmis tarkastusmetodi käy läpi asennettuja liitännäisiä, joihin kuuluvat mm. PHP ja MySQL, ja metodi tarkisti jokaisen versionumeron, että ne ovat tarpeeksi uusia versioita. Lisäosataulukon yläpuolella on yhteenveto siitä, täyttävätkö järjestelmän ohjelmistokehyksen vaatimat minimivaatimukset.

Ohjelmistokehys sisältää oletuksena yhden testiprojektin, jota pääsee selaimen kautta kokeilemaan. Selaimen annetaan koneen osoite, joka avaa testisivun. Sivun sisältää neljä välilehteä, joissa opastetaan, mistä löytyvät ne tiedostot, joiden kautta pääsee muokkaamaan kyseistä sivua. Yksi välilehti sisältää valmiin yhteyslomakkeen, johon on lisätty kenttien tarkastus ja kuvallinen varmistus lähetys-painikkeen yhteyteen, että kyseessä ei ole tietokone, vaan ihminen. Viimeinen välilehti sisältää kirjautumislomakkeen, jonka kautta pääsee kokeilemaan käyttäjänhallintaa. Hallinnan kautta voidaan lisätä, poistaa ja muokata käyttäjiä.

## 7.3 Ympäristömuuttujat ja uusi projekti

Windows-järjestelmään voi asettaa ympäristömuuttujat helpottamaan komentorivillä työskentelyä. Tässä projektissa ne lisätään, jotta pystyin käyttämään Yiiin komentorivityökaluja. Lisättävät polut ovat PHP:n tiedostot sisältävä hakemisto ja Yiiin hakemisto,

joka sisältää komentorivisovellukset. Nämä kaksi kohtaa lisätään path-muuttujaan. Muuttujien asettamisen jälkeen pitää kaikki komentorivit käynnistää uudelleen, jotta muutokset tulevat voimaan. Toiminnan pystyy varmistamaan avaamalla uuden komentorivin ja syöttämällä sinne sovelluksia, jotka löytyivät polkuun lisätyistä hakemistoista.

Ympäristömuuttujien asettamisen jälkeen pääsee kokeilemaan uuden projektin luomista. Komentorivityökalu, jolla voidaan luoda uusia projekteja, on *yiic*. Uuden projektin saa luotua siirtymällä komentorivillä palvelimen WWW-hakemistoon ja suorittamalla siellä komennon *yiic webapp [projektin\_nimi]*. Enterillä hyväksymisen jälkeen ohjelma kysyy, halutaanko varmasti luoda tarvittavat tiedostot uuteen projektihakemistoon. Hyväksyminen tähän onnistuu kirjoittamalla komentoriville *yes* ja hyväksymällä tämä enteriä painamalla. Hyväksymisen jälkeen ohjelma generoi tarvittavat tiedostot projektin nimen mukaiseen hakemistoon WWW-palvelimen juurihakemistoon. Selaimen kautta pääsee tarkastamaan juuri luodun projektin.

Luodussa hakemistossa on rakenne valmiina, mihin lisätään haluttuja tiedostoja ja luokkia tai muokataan jo valmiiden luokkien metodeita. Projektin juurihakemisto sisältää *index.php* -tiedoston, joka on sivuston komentosarjatiedosto. Juurihakemistosta löytyy hakemisto *protected*, joka sisältää sovelluksen suojatut tiedostot. Näihin lukeutuu mm. komentorivisovellukset, komponentit, kokoonpanotiedostot, käsittelijät, tietokantamallit, laajennukset ja luokkatiedostot. Enemmän aiheesta löytyy oppaasta (kts. Liite 1).

## 8 POHDINTA

Projekti lähti liikkeelle kiinnostuksesta WWW-tekniikoihin ja halusta oppia jotain, mitä ei ollut vielä ennen ollut osannut. Aiempaa kokemusta ohjelmistokehyksistä ei juuri ollut ennen projektin aloittamista. Aluksi ajatuksena oli suunnitella virtuaalikirssi kyseisestä aiheesta ja tehdä materiaalit ynnä muut tarvittavat kurssia varten. Tämä alkoi kuitenkin tuntua tiedonhaun aikana paljon suuremmalta urakalta kuin olisi aluksi osannutkaan kuvitella. Yllättävän paljon olisi ollut työtä virtuaalikirssin eteen, varsinkin kun aiempaa kokemusta ei ole. Tämän jälkeen ajatus virtuaalikirssin toteuttamisesta jäi taka-alalle, mutta halusin silti pysyä saman aiheen äärellä. Tässä vaiheessa joku ehdotti-kin jo, että aiheesta saisi hyvän oppaan ja voisin keskittyä enemmän ohjelmistokehykseen ja saada siitä enemmän irti.

Idea meni eteenpäin siitä, mitä opinnäytetyössäni käsittelisin. Opas alkoi kuulostaa realistiselta, ja päätin lyödä aiheen lukkoon. Samalla tuli haettua tietoa eri lähteistä ja tutustuttua aiheeseen paremmin ennen varsinaisen työn aloittamista. Oppaan tekemiseen oli hyvin vähän tarjolla aineistoa. Tässä kohtaa piti soveltaa ja hakea tietoa aiheen vierestä. Sopivia aiheita löytyi mm. nettijulkaisuja käsittelevistä kirjoista ja esityksiin tarkoitettuista kirjoista. Jälkimmäistä luettaessa tuli mieleen, että tämä hyödyttää myös opinnäytetyön seminaarin yhteydessä pidettävän esitelmän suunnittelua ja toteutusta

Eri ohjelmistokehyksistä löytyi tietoa vaihtelevasti. Joistain oli kirjoitettu erikseen kirja ja nettijulkaisuja. Myös useassa tapauksessa oli dokumentointi hyödynnettävissä ja yhteisö, joka neuvoi tarvittaessa ja, mistä löytyi tietoa ongelmien varalle. Ohjelmistokehys, johon tuli itse päädyttyä, sisältää hyvän dokumentoinnin, kirjoitettua aineistoa ja nettiyhteisön tuen. Tässä kohtaa tietoa oli jo liikaakin ja täytyi aloittaa rajaaminen siitä, mitä ottaa mukaan, ja mitä ei.

Oppaan tekeminen eteni samaan tahtiin, kun kävin läpi ja kokeilin eri toiminnallisuuksia Yii-ohjelmistokehysten kanssa. Tämä taktiikka toimi aika hyvin, mutta toisaalta olisi voinut käydä aina yhden kokonaisuuden kerrallaan läpi ja vasta sitten kirjoittaa, mitä on tullut tehtyä. Lopputulos oppaan kannalta ei ole ehkä selkeimmästä päästä. Ja se ei sovi välttämättä ihan aloittelevalle henkilölle, mutta jo vähän asiasta tietävälle se so-



pii jo paremmin. Sekavuutta oppaaseen toi alueen rajaus, joka ei ollut alussa täysin selvillä ja matkan varrella se alkoi vasta hahmottua.

Jos opas menisi jatkokehitykseen, tulisi se suunnitella uudestaan. Ulkoasun muotoiluun voisi panostaa enemmän ja käyttää kuvia ja tekstiä tehokkaammin keskenään. Opas vaatii selkeyttämistä ja rakenteen voisi myös suunnitella kokonaan uudestaan. Alussa sain idean, että yhdellä sivulla olisi yksi käsitelty asia ja se sisältäisi kuvan tekstin kera. Tekstiä tuli kuitenkin enemmän ja kuvia suhteessa siihen oli vähemmän. Asennusohjeet tuli tehtyä vain yhteen järjestelmään, joten jos haluaisi jatkaa projektin parissa, niin ohjeet tulisi tehdä myös useammalle järjestelmälle. Niitä ovat mm. Unix ja Mac.

Oppaan voisi siirtää WWW-palvelimelle, mikä mahdollistaisi sille moninkertaisesti erilaisia mahdollisuuksia ja osaltaan helpottaisivat luettavuutta ja selkeyttäisi paljon ulkoasua. Selkeyttä saisi tuomaan kuvien pienentäminen, ja ne avautuisivat tarvittaessa. Koodiesimerkit saisi näkymään muotoiltuina ja niitä pystyisi kopioimaan suoraan sivuilta testattavaksi. Päivittäminen olisi hieman kätevämpää myös nettijulkaisussa ja samassa yhteydessä voisi olla kommentointimahdollisuus. Tämä antaisi mahdollisuuden käyttäjille kommentoida opasta, kertomaan mahdolliset epäkohdat ja uudet ideat. Ehkä siinä olisi jatkokehitysideaa taas hetkeksi.

## LÄHTEET

Alter Way. 2013. Wampserver. Luettu 16.4.2013.  
<http://www.wampserver.com/en/>

Av-viestintä. 2011. Luettu 18.4.2013.  
<http://www.avkymppi.net/joomla/wamp.html>

Cake Software Foundation. 2013. Why use CakePHP. Luettu 10.4.2013.  
<http://cakephp.org/>

Don Hon. 2013. Notepad++. Luettu 10.4.2013.  
<http://notepad-plus-plus.org/>

Ellis Lab Inc. 2013a. About EllisLab. Luettu 18.4.2013.  
<http://ellislab.com/about/>

Ellis Lab Inc. 2013b. CodeIgniter. Luettu 12.4.2013.  
<http://ellislab.com/codeigniter/>

Fabien Potencier. 2013a. About. Luettu 13.3.2013.  
<http://symfony.com/about/>

Fabien Potencier. 2013b. Symfony. Luettu 13.3.2013.  
<http://symfony.com/>

James, W. Grenning. Test-Driven Development for Embedded C. 2011. The Pragmatic Bookself. Raleigh, North Carolina – Dallas, Texas

Jamsi. 2013. How to Bootstrap PHP Code. Luettu 17.4.2013.  
<http://www.jigsawboys.com/2009/09/17/how-to-bootstrap-php-code/>

Kai S. 2013. XAMPP. Luettu 18.4.2013.  
<http://www.apachefriends.org/en/xampp.html>

Keränen, V. 2007. Verkko-oppimateriaalin tuottajan opas. Docendo. Jyväskylä.

Keränen, V, Lamberg N, Penttinen J. 2006. Web-julkaiseminen & multimedia. Docendo. Jyväskylä.

Lammi, O. 2009. Vaikuta visuaalisesti! Laadi selkeä esitys. Docendo. Jyväskylä.

Laurent, A, Emmanuel F, Thierry M. 2013a. EasyPHP. Luettu 17.4.2013.  
<http://www.easyphp.org/>

Laurent, A, Emmanuel F, Thierry M. 2013b. Introduction. Luettu 17.4.2013.  
<http://www.easyphp.org/introduction.php>

Makarov, A. 2011. Yii 1.1 Application Development Cookbook. Packt Publishing, Birmingham – Mumbai.

Nielsen, J. 2000. WWW-suunnittelu. Edita. Jyväskylä.

Okanovic, V. 2011 Designing a new web application framework. Luettu 14.4.2013. Fac. of Electr. Eng., Comput. Sci. Dept., Univ. of Sarajevo, Sarajevo, Bosnia-Herzegovina.

Oracle Corporation. 2013. VirtualBox. Luettu 10.4.2013.  
<https://www.virtualbox.org/>

PHP Frameworks. 2013. Top 10 Ranking PHP Frameworks. Luettu 15.5.2013.  
<http://www.phpframeworks.com/top-10-php-frameworks/>

Sanoma News Oy. 2009. Windows 7 myyntiin lokakuussa. Luettu 10.4.2013.  
<http://www.digitoday.fi/bisnes/2009/06/03/windows-7-myyntiin-lokakuussa/200913667/66>

Softonic International S.L. 2013. XAMPP. Luettu 17.4.2013.  
<http://xampp.en.softonic.com/>

The Register. 2008. Sun nabs innotek's 20MB of open source, virtualized goodness.  
[https://www.theregister.co.uk/2008/02/12/sun\\_innotek\\_virtualization/](https://www.theregister.co.uk/2008/02/12/sun_innotek_virtualization/)

Winesett, J. 2010. Agile Web Application Development with Yii 1.1 and PHP5. Packt Publishing, Birmingham – Mumbai.

Yii Software LLC. 2013a. Downloads. Luettu 15.4.2013.  
<http://yii.googlecode.com/files/yii-docs-1.1.13.e9e4a0.zip>

Yii Software LLC. 2013b. Luettu 12.4.2013.  
<http://www.yiiframework.com/>

Yii Software LLC. 2013c. Model-View-Controller. Luettu 18.4.2013.  
<http://www.yiiframework.com/doc/guide/1.1/en/basics.mvc>

Zend Technologies Ltd. 2013a. Zend framework. Luettu 14.4.2013.  
<http://framework.zend.com/>

Zend Technologies Ltd. 2013b. About. Luettu 14.4.2013.  
<http://framework.zend.com/about/>

## **LIITTEET**

Liite 1. PHP Framework Yii Opas



yiiframework

Opas

<b>Johdanto</b> .....	<b>4</b>
<b>Yii Framework</b> .....	<b>5</b>
Vaatimukset.....	5
MVC-Malli Yii:ssä .....	5
<b>Työympäristön asennus</b> .....	<b>9</b>
Windows.....	9
<b>Yii Frameworkin asennus</b> .....	<b>11</b>
Windows 7 .....	11
<b>Uuden projektin luominen</b> .....	<b>12</b>
Hakemistorakenne .....	13
Konehuone .....	14
Setters and getters .....	14
Yiiin tapahtumat .....	16
Tuonti ja tarvelataus.....	17
Poikkeukset.....	18
Komponenttien konfigurointi .....	18
Pyynnöt (request).....	20
Hello World.....	20
Reititetty pyyntö.....	21
Dynaamisen sisällön lisääminen .....	21
Sivujen linkittäminen .....	22
Yhdistäminen tietokantaan .....	22
CRUD operaatioiden toteuttaminen .....	23
Gii code generator .....	23
TDD-kehitys .....	24
Asennus .....	24
TDD-kehitys .....	27
Lomakkeen tarkistus.....	27

CAPTCHA.....	29
<b>Muuta.....</b>	<b>33</b>
HTML-toiminnot .....	33
Pudotusvalikko .....	33
Linkkien luonti .....	34

## Johdanto

Tämä opas on tehty osana opinnäytetyönäni. Sen tarkoituksena on tuoda materiaalia kaikkien aiheesta kiinnostuneiden käytettäväksi. Oppaassa käydään läpi työympäristön asennus ja alkeet ohjelmistokehitys Yleistä. Aluksi on tekstiä yleisesti Yii-ohjelmistokehityksestä ja sen toimintatavasta. Tämän jälkeen asennetaan työympäristö Windows-käyttöjärjestelmään, asennetaan kaikki tarvittavat paketit ja laitetaan ympäristömuuttajat kuntoon. Asiat on pyritty pitämään selkeinä ja helppolukuisina. Tämä työ sisältää osittain samoja asioita, mitä raportissa on, joka on kirjoitettu tämän työn pohjalta.



## Yii Framework

Yii Framework -projekti on aloitettu alun perin tammikuussa vuonna 2008 Qiang Xuen toimesta. Sen tarkoitus on alusta asti ollut avoimeen lähdekoodiin perustuva ohjelmistokehitysprojekti. Ensimmäinen alkeisversio tuli lokakuussa vuonna 2008 ja versio 1.0 julkaistiin joulukuussa samana vuonna. Nimi Yii tulee sanoista *yes, it is*. Tämä tarkoittaa helppoa, tehokasta ja laajennettavaa.

Yii on tehokas komponenttipohjainen WWW-ohjelmistokehys, joka on kirjoitettu PHP5-ohjelmointikielellä. Sen tarkoituksena on luoda niin pieniä kuin suuriakin WWW-sovelluksia ja sivustoja. Laajennettavuutta tuo valmiit liitännäiset, joita pystyy myös itse tekemään ja liittämään yhteen tai käyttämään niitä useissa projekteissa. Yii ohjelmistokehys sisältää useita funktioita, joiden on tarkoitus helpottaa ja nopeuttaa työskentelyä. Nämä funktiot ovat ryhmitelty luokiksi.

Toimintaan tarvitaan jokin käyttöympäristö, mistä löytyy tuki PHP5:lle, MySQL:lle ja APACHE:lle. Tässä oppaassa on käytetty Windows-ympäristöä, johon on asennettu kaikki tarvittavat osat, jotta pystytään pyörittämään ohjelmistokehystä. Oppaan ohjelmointiosuuksia voi hyödyntää myös muiden järjestelmien kanssa. Asennusohjeet ovat kuitenkin vain Windows 7-käyttöjärjestelmään.

## Vaatimukset

- WWW-palvelinohjelmisto, joka tukee PHP 5.1.0 tai uudempaa.
- Vaihtoehtoisia tietokannoiksi MySQL ja SQLite.
- Ymmärrystä oliopohjaisesta ohjelmoinnista on etu.

## MVC-Malli Yii:ssä

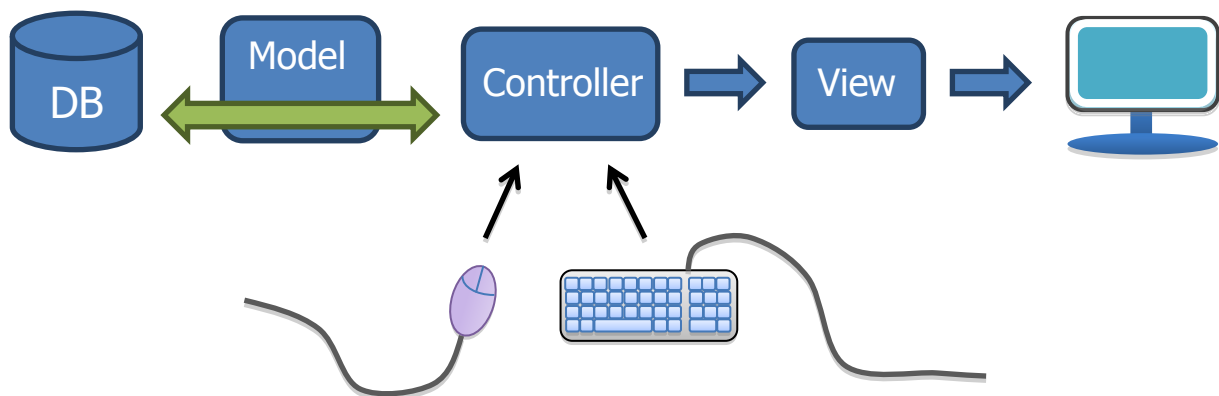
MVC-mallin avulla saadaan eroteltua käyttöliittymä sovellusalueiedoista. MVC-malli jakautuu kolmeen osaan, jotka ovat malli, näkymä ja käsittelijä.

**Mallin** (Model) tehtävänä on ylläpitää tilaa ja määrittää tiedon väliset suhteet tietokantatauluissa. Se hoitaa myös kirjoittamisen ja lukemisen tietokantaan. Jokaisella tietokantataululla on oma mallinsa ja ne ovat nimetty tietokantataulun mukaan. Mallin toimintaan saamiseksi ei tarvitse välttämättä kirjoittaa yhtään koodia. Sen saa generoitua valmiina käyttäen komentorivityökalua yiic tai selaimessa yii-moduulia. Mallilla on kaksi tärkeätä tehtävää, jotka ovat etsiä ja tallentaa tietoa. Tietokantahakuun ei tarvitse kirjoittaa erillisiä hakuja, vaan ne onnistuvat hyvin yksinkertaisella haululla. Jos taulusta halutaan hakea kaikki tieto, se käy lauseella `ModelName.FindAll`. (Winesett, J. 2010, 10.)

**Näkymä** (View) vastaa tyypillisesti käyttöliittymän muodostamisesta, joka perustuu tietoon mallissa. Näkymä on PHP-komentosarja, joka sisältää käyttöliittymään liittyviä elementtejä, jotka ovat yleensä rakennettu HTML:ää käyttäen, mutta ne voivat sisältää myös PHP-lauseita. Nämä PHP-lauseet ovat yleensä hyvin yksinkertaisia ehtoja tai silmukoita, jotka viittaavat muihin Yiin elementteihin, kuten HTML-apuluokkaan tai esikäännettyihin ohjelmakomponentteihin. Näkymä saa tietoja käsittelijältä ja näyttää ne käyttäjälle. Tämä osio on ainoa, jossa käytetään myös HTML-, CSS- ja JavaScript- ym. kieliä. (Winesett, J. 2010, 10.)

**Käsittelijä** (Controller) ohjaa reititetyt pyynnöt ja on vastuussa käyttäjän syötteistä, vuorovaikutuksessa mallin kanssa ja päivittää näkymää tarpeen mukaan. Käsittelijä on esiintymä CController-luokasta tai sen lapsiluokasta. Kun käsittelijä saa toiminnan, suorittaa se pyydetyt tehtävät, jotka ovat vuorovaikutuksessa tarvittaviin malleihin, jonka jälkeen se muodostaa näkymän. Käsittelijää kutsutaan aina kun annetaan osoiteriville uusi osoite. Se voi lukea tiedot POST-, GET- ja SESSION-muuttujista ja käsitellä tietokantoja. MVC-arkkitehtuurin kanssa osoitteilla on tietty rakenne. Ne sisältävät parametrin, mitä käsittelijää tullaan käyttämään. Kaikki navigointi hoidetaan parametrin ja käsittelijän kautta. (Winesett, J. 2010, 10.)

[www.example.com/index.php?r=myController](http://www.example.com/index.php?r=myController)

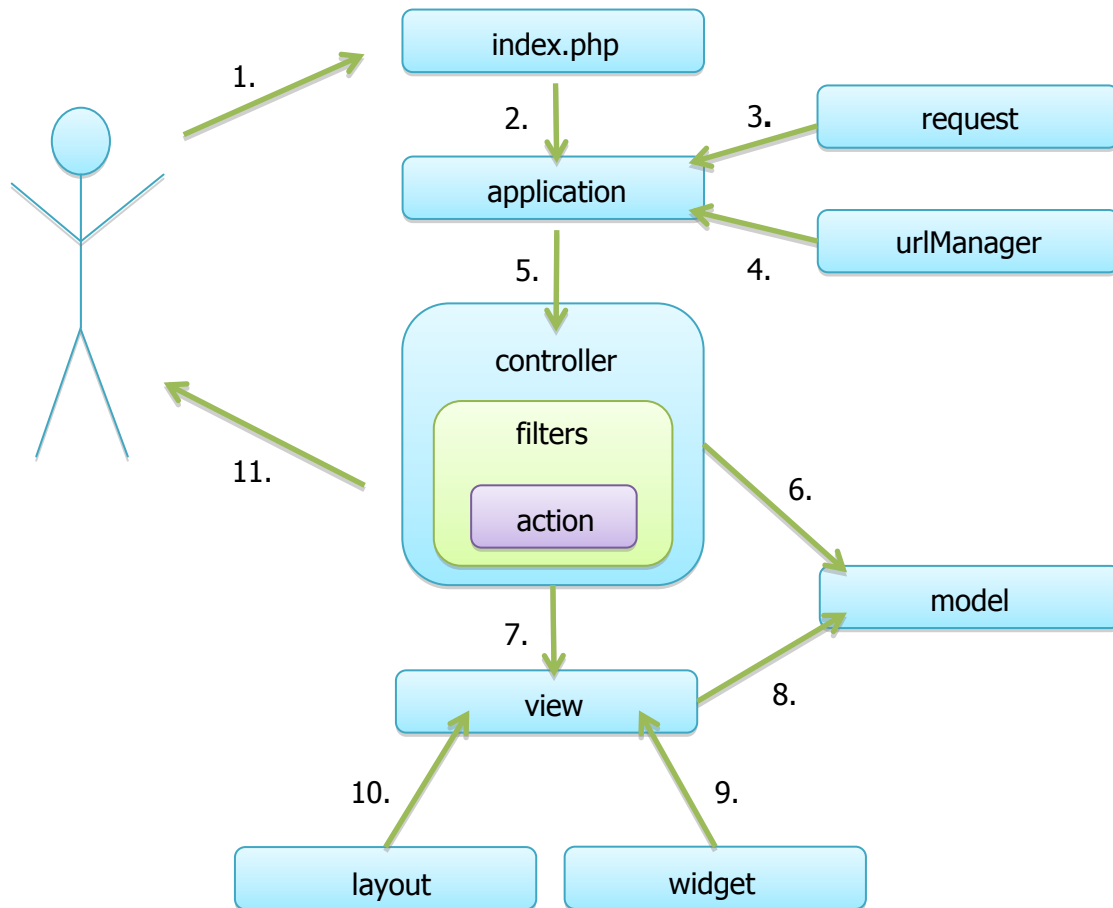


**Action** on käsittelijän toiminta, ja niitä voi olla käsittelijässä useita. Jokaisella toiminnolla on oma toiminto. Käyttäjien hallintaan tarkoitettu käsittelijä voi sisältää toimintoja esimerkiksi käyttäjän lisäämiseen, poistamiseen ja muokkaamiseen. Jos toimintoa ei aseteta, käytetään oletustoimintoa, joka on käsittelijässä `actionIndex()`-metodi. Alla olevassa linkissä esimerkki toiminnan käyttämisestä:

[www.example.com/index.php?r=myController/myAction](http://www.example.com/index.php?r=myController/myAction)

Jatkossa tässä oppaassa käytetään yllä olevia termejä seuraavasti:

- Model                      Malli
- View                        Näkymä
- Controller                Käsittelijä
- Action                     Tapahtuma
- Framework                Ohjelmistokehys



1. Käyttäjä avaa sivun <http://www.example.com/index.php?r=post/show&id=1> ja WWW-palvelin käsittelee pyynnön *bootstrap*-komentosarjan, joka sijaitsee *index.php*-tiedostossa.
2. Bootstrap komentosarja luo Application-luokan istunnon ja suorittaa sen.
3. Application-luokan istunto pitää hallussaan yksityiskohtaista tietoa käyttäjän pyytämästä tiedosta, jonka se saa Application-komponentilta, joka on nimetty *request*.
4. Application-luokan istunto määrittää käsittelijän ja tapahtuman pyynnöt sovelluksen komponentilla *urlManager*.
5. Application luo istunnon pyydetyistä ohjaimista myöhempiä käyttäjän pyyntöä varten. Käsittelijä määrittää toiminnon *show*, joka viittaa metodiin *actionShow*, joka sijaitsee Käsittelijä-luokassa. Sen jälkeen käsittelijä luo ja suorittaa suotimet (esim. pääsyn hallinta, benchmarkkaus), jotka ovat yhdistettynä tähän toimintoon.
6. Toiminto lukee Post-mallin tietokannasta, jonka ID on 1.
7. Toiminto muodostaa näkymän, jonka nimi on *show* ja se on Post-mallin mukana.
8. Näkymä lukee ja näyttää Post-mallin attribuutit.
9. Näkymä suorittaa tarvittavat ohjelmakomponentit.
10. Näkymän muodostamisen tulos on upotettu sivupohjaan.
11. Toiminto suorittaa loppuun näkymän muodostamisen ja näyttää tuloksen käyttäjälle.

## Työympäristön asennus

### Windows

Tässä osiossa käydään läpi työympäristön asennus Windows 7 -käyttöjärjestelmään. Läpikäytävät kohteet ovat verkkosovellusten kehitysympäristön asennus ja ympäristömuuttujien asettaminen. Erikseen ladattavan virtuaalipalvelimen yhteydessä on kaikki tarvittavat osat, mitä Yii Framework tarvitsee toimiakseen. Nämä osat ovat APACHE, PHP ja MySQL, jotka tulevat Wampserverin paketin mukana. Mahdollista on käyttää myös jotain muuta palvelinohjelmistoa, joka tukee PHP versiota 5.1.0 tai uudempaa. Asennuspaketti löytyy kohdasta:

- <http://www.wampserver.com/en/>

Tarjolla on useampia versioita niin 32bit- ja 64bit- käyttöjärjestelmille. Asennuspaketin voi valita oman järjestelmän vaatimusten mukaan, mutta jos ei ole varma, niin kannattaa valita 32bit- asennustiedosto. Asennus on tehty helpoksi ja se ei tarvitse mitään erityisiä toimia. Kunhan vain käy asennuksen läpi. Muitakin vaihtoehtoja WWW-palvelimille on tarjolla, mutta tässä oppaassa käydään läpi vain Wampserver.

Miksi juuri Wampserver? Se on helppokäyttöinen ja alkuun pääsee hyvin tekemättä mitään muutoksia asetuksiin. Paketissa tulee kaikki tarvittava, jotta saa toimivan virtuaalipalvelimen pystyyn. Mahdollista on käyttää myös eri tietokantoja. Huonona puolena voi pitää virallisen asennusohjeen puuttumista. Alla linkki keskeneräiseen opukseen, joka kirjoitushetkellä oli vielä kirjoitusvaiheessa:

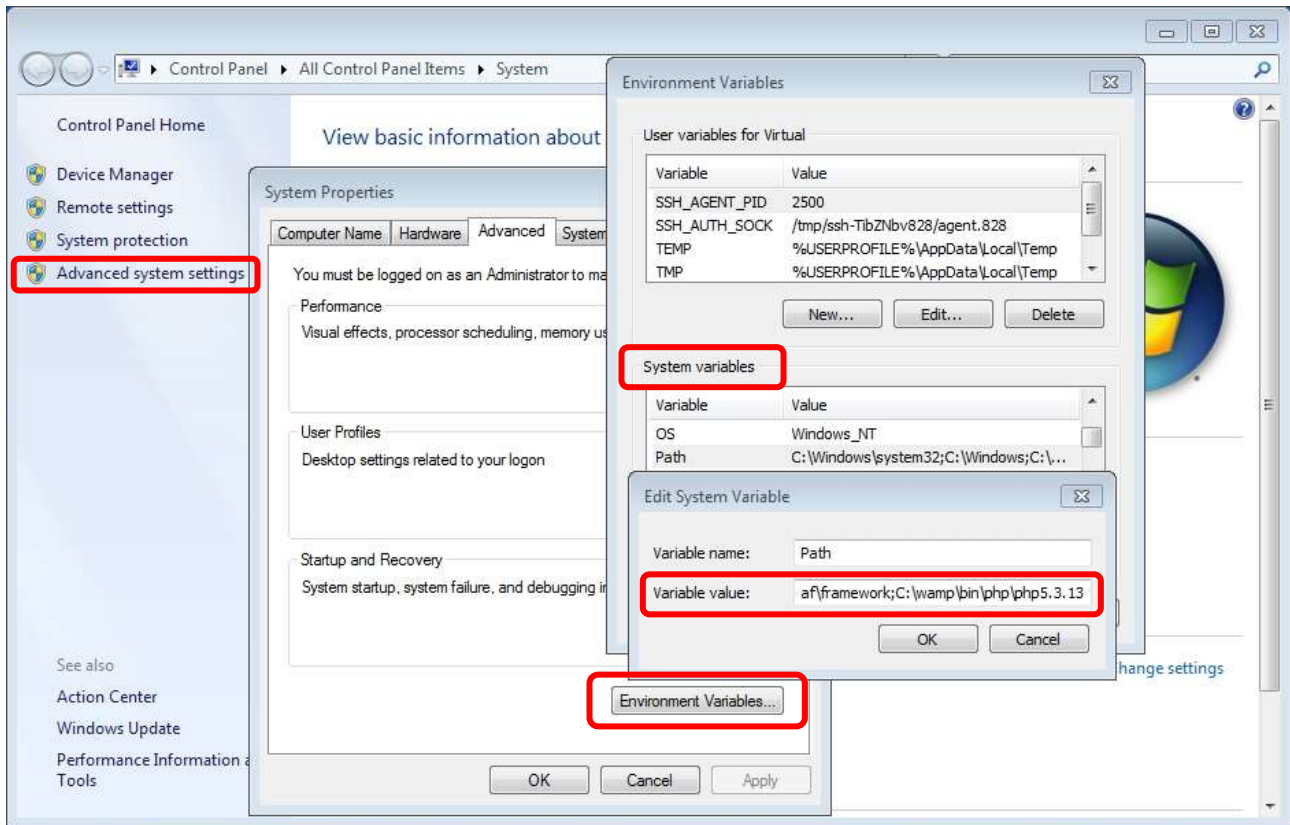
- <http://forum.wampserver.com/read.php?2,87317,87338#msg-87338>

Wampserver voi vaatia toimiakseen Visual C++ 2010 SP1-paketin. Tästä on tarjolla vain 32bit- versio. Jos palvelin ei lähde toimimaan asennuksen jälkeen, voi seuraavan paketin asentaa ja kokeilla uudestaan:

- <http://www.microsoft.com/download/en/details.aspx?id=8328>

Kun Wampserver ja Visual C++ 2010 SP1 on asennettu, on seuraavana asetettava ympäristömuuttujat kuntoon. Näitä tarvitaan jatkossa, jos halutaan käyttää automaattisia työkaluja koodin generoimiseen ja mm. uuden sivupohjan luomiseen käyttäen komentorivityökalua. Ympäristömuuttujiin pääsee esimerkiksi näppäinyhdistelmällä *'Windows Logo+Break'* tai työpöydältä painamalla tietokoneen kuvakkeen päällä hiiren kakkosnäppäimellä ja valitsemalla *'properties'*. Aukeavan ikkunan vasemmassa laidassa on paneeli, josta löytyy *'Advanced system settings'* -linkki. Linkistä aukeaa ikkuna, jossa pitäisi olla välilehti *Advanced* valittuna. Tästä näkymästä valitaan painike *'Environment Variables'*. Polut asetetaan *'System variables'* - otsikon alle olevaan tekstilaatikkoon. Seuraavat polut lisätään puolipisteellä erotettuina:

- Path-muuttujaan lisätään PHP5:n hakemistopolku ja Yii frameworkin framework-hakemisto.



Kun ympäristömuuttujat on asetettu, osaa ohjelmistokehys käyttää niitä asetetuista poluista. Näitä käytetään eri komentojen ajamiseen ja framework-hakemistosta löytyvät mm. Yiiin komentorivityökalut. Ympäristömuuttujien tallentamisen jälkeen kannattaa sulkea avoinna olevat komentorivit, sillä muutokset eivät ole tulleet voimaan kyseiseen ikkunaan.

## Yii Frameworkin asennus

Tässä osiossa käydään läpi asennus vaihe vaiheelta. Lopuksi tarkistetaan järjestelmän vähimmäisvaatimukset.

### Windows 7

Tässä oppaassa käytetään versiota 1.1.2, joten on suositeltavaa käyttää samaa versiota. Yiin ohjelmakoodi löytyy linkistä *'download archived releases'*:

- <http://www.yiiframework.com/download/> [5.3.2013]

Kirjoitushetkellä paketti oli nimetty yii-1.1.2.r2086p.zip, joka puretaan Wampserverin WWW-hakemistoon. Jos Wampserver ei ole käynnissä, käynnistä se nyt. Tämän jälkeen selaimen avataan osoite <http://localhost> tai <http://127.0.0.1>, joka avaa Wampserverin aloitusnäkyvän. *'Your Projects'* -otsikon alta pitäisi nyt löytyä hakemisto, joka on nimetty ladatun paketin tapaisesti. Valitsemalla selaimesta kyseisestä hakemistosta *'requirements'*-hakemisto, saadaan näin auki uusi sivu. Tämä sivu tarkistaa täyttääkö palvelimen kokoonpano Yiin minimivaatimukset. Kaikki on kunnossa, jos taulukon yläpuolella lukee Conclusion-otsikon alla *'Your server configuration satisfies the minium requirements by Yii'*.

## Yii Requirement Checker

### Description

This script checks if your server configuration meets the requirements for running [Yii](#) Web applications. It checks if the server is running the right version of PHP, if appropriate PHP extensions have been loaded, and if php.ini file settings are correct.

### Conclusion

Your server configuration satisfies the minimum requirements by Yii. Please pay attention to the warnings listed below if your application will use the corresponding features.

### Details

Name	Result	Required By	Memo
PHP version	Passed	<a href="#">Yii Framework</a>	PHP 5.1.0 or higher is required.
\$_SERVER variable	Passed	<a href="#">Yii Framework</a>	
Reflection extension	Passed	<a href="#">Yii Framework</a>	
PCRE extension	Passed	<a href="#">Yii Framework</a>	
SPL extension	Passed	<a href="#">Yii Framework</a>	
DOM extension	Passed	<a href="#">CWsdGenerator</a>	
PDO extension	Passed	All <a href="#">DB-related classes</a>	
PDO SQLite extension	Passed	All <a href="#">DB-related classes</a>	This is required if you are using SQLite database.
PDO MySQL extension	Passed	All <a href="#">DB-related classes</a>	This is required if you are using MySQL database.
PDO PostgreSQL extension	Warning	All <a href="#">DB-related classes</a>	This is required if you are using PostgreSQL database.
Memcache extension	Warning	<a href="#">CMemCache</a>	
APC extension	Warning	<a href="#">CApcCache</a>	
Mcrypt extension	Passed	<a href="#">CSecurityManager</a>	This is required by encrypt and decrypt methods.
SOAP extension	Warning	<a href="#">CWebService</a> , <a href="#">CWebServiceAction</a>	
GD extension	Passed	<a href="#">CCaptchaAction</a>	

■ passed
 ■ failed
 ■ warning

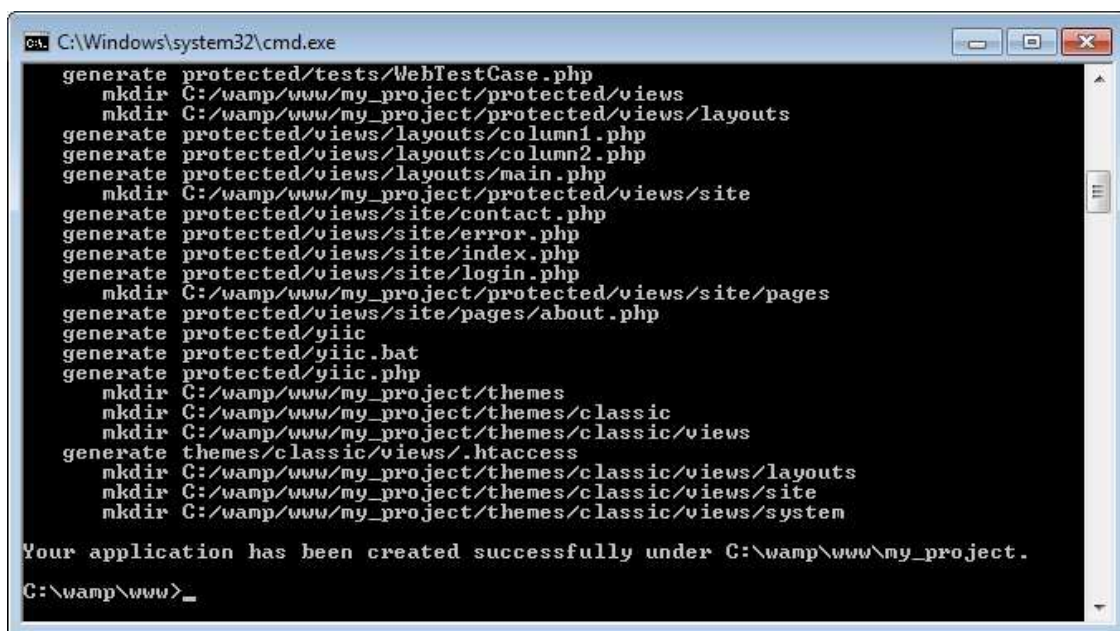
## Uuden projektin luominen

Uuden projektin luomiseen voidaan käyttää yiic-komentorivityökalua, joka generoi automaattisesti tarvittavat hakemistot ja tiedostot. Tämän tekemisessä on tärkeää, että ympäristömuuttujat ovat asetettu Windows -ympäristöön oikein. Muuten sivut eivät välttämättä generoidu tai toimi oikein. Hakemistorakenteen ja tarvittavat luokat voidaan tehdä myös manuaalisesti.

Avaa komentorivi ja navigoi Wampserverin WWW-hakemistoon. Uuden sivupohjan saa luotua komennolla:

- `Yiin_hakemisto\framework\yiic webapp [projektin_nimi]`
- `c:\wamp\www\framework\yiic webapp my_project`

Hyväksy valinta kirjoittamalla 'Y' ja paina enter.



```

C:\Windows\system32\cmd.exe
generate protected/tests/WebTestCase.php
mkdir C:/wamp/www/my_project/protected/views
mkdir C:/wamp/www/my_project/protected/views/layouts
generate protected/views/layouts/column1.php
generate protected/views/layouts/column2.php
generate protected/views/layouts/main.php
mkdir C:/wamp/www/my_project/protected/views/site
generate protected/views/site/contact.php
generate protected/views/site/error.php
generate protected/views/site/index.php
generate protected/views/site/login.php
mkdir C:/wamp/www/my_project/protected/views/site/pages
generate protected/views/site/pages/about.php
generate protected/yiic
generate protected/yiic.bat
generate protected/yiic.php
mkdir C:/wamp/www/my_project/themes
mkdir C:/wamp/www/my_project/themes/classic
mkdir C:/wamp/www/my_project/themes/classic/views
generate themes/classic/views/.htaccess
mkdir C:/wamp/www/my_project/themes/classic/views/layouts
mkdir C:/wamp/www/my_project/themes/classic/views/site
mkdir C:/wamp/www/my_project/themes/classic/views/system

Your application has been created successfully under C:\wamp\www\my_project.
C:\wamp\www>_
  
```

Ohjelma generoi tiedostot ja ne löytyvät tämän jälkeen Wampserverin WWW-hakemistosta antamallasi projektin nimellä. Luotua sivupohjaa pääsee kokeilemaan kohdassa [http://localhost/my\\_project](http://localhost/my_project) joka sisältää muutamia valmiita sivuja.



## Hakemistorakenne

Hakemistojen ja tiedostojen tehtävät generoidussa sivupohjassa ovat seuraavia:

### my\_project/

index.php	sivuston komentosarjatiedosto
assets/	sisältää julkaistut resurssitiedostot
css/	sisältää CSS tiedostot
images/	sisältää kuvatiedostot
themes/	sisältää sovelluksen teemat
protected/	sisältää sovelluksen suojatut tiedostot
yiic	komentorivisovellus Unix/Linux
yiic.bat	komentorivisovellus Windows
yiic.php	komentorivisovellus PHP
commands/	sisältää muokatut 'yiic'-käskyt
shell/	sisältää muokatut 'yiic shell'-komennot
components/	sisältää uudelleenkäytettävät komponentit
Controller.php	kantaluokka Controller-luokille
UserIdentity.php	luokka käyttäjien todennukseen
config/	sisältää kokoonpanotiedostot
console.php	konsolin kokoonpanotiedosto
main.php	web-ohjelmiston kokoonpanotiedosto
test.php	kokoonpanotiedosto toiminnalliselle testaukselle
controllers/	sisältää controller-luokat
SiteController.php	oletusarvoinen controller-luokka
data/	sisältää esimerkkiluokat tietokannoista
schema.mysql.sql	tietokantamalli MySQL-tietokannalle
schema.sqlite.sql	tietokantamalli SQLite-tietokannalle
testdrive.db	SQLite-tietokannan mallitiedosto
extensions/	sisältää kolmannen osapuolen liitännäiset
messages/	sisältää käännetyt viestit
models/	sisältää mallien luokkatiedostot
LoginForm.php	malli kirjautumistapahtumalle
ContactForm.php	malli yhteydenottotapahtumalle
runtime/	sisältää väliaikaistiedostoja

tests/	sisältää testaus komentosarjoja
views/	sisältää controller-luokan näkymät ja leiskat
layouts/	sisältää leiskat
main.php	oletusarvoinen leiska joka sivulle
column1.php	leiska yksi sarakkeisille sivuille
column2.php	leiska kaksi sarakkeisille sivuille
site/	sisältää näkymät site-käsittelijälle
pages/	sisältää staattiset sivut
about.php	näkymä about.php-tiedostolle
contact.php	näkymä contact-toiminnalle
error.php	näkymä virhesivuille
index.php	näkymä index-toiminnalle
login.php	näkymä login toiminnalle

## Konehuone

### Setters and getters

Kuinka määrittää omia luokan ominaisuuksia käyttäen gettereritä ja settereitä.

1. PHP:ssa ei ole ominaisuuksia koodin tasolla, joten käytetään gettereitä ja settereitä seuraavalla tavalla:

```
class MyClass
{
    // piilotettu $property
    private $property;
    // getter
    public function getProperty()
    {
        return $this->property;
    }
    // setter
    public function setProperty($value)
    {
        $this->property = $value;
    }
}
$object = new MyClass();
// asetetaan arvo
$object->setProperty('value');
// haetaan arvo
echo $object->getProperty();
```

2. Kutsutaan gettereitä ja settereitä luokan jäseninä:

```
// CComponent luokka peritään
class MyClass extends CComponent
{
    private $property;

    public function getProperty()
    {
        return $this->property;
    }

    public function setProperty($value)
    {
        $this->property = $value;
    }
}

$object = new MyClass();
$object->property = 'value'; // sama kuin $object->
setProperty('value');
echo $object->property; // sama kuin $object->getProperty();
```

3. Seuraavalla tavalla saa ominaisuuksista vain luku-, tai vain kirjoitus -muotoisia:

```
class MyClass extends CComponent
{
    private $read = 'ainoastaan luku';
    private $write = 'ainoastaan kirjoitus';

    public function getRead()
    {
        return $this->read;
    }

    public function setWrite($value)
    {
        $this->write = $value;
    }
}

$object = new MyClass();
// gives us an error since we are trying to write to read-only
property
$object->read = 'value';
// echoes 'read only property'
echo $object->read;
// gives us an error since we are trying to read to write-only
property
echo $object->write;
// writes 'value' to private $write
$object->write = 'value';
```

4. Yii käyttää tätä tekniikkaa, koska lähes kaikki on komponentteja. Kutsuttaessa esimerkiksi

```
Yii::app()->user->id
```

saadaan tällä nykyisen kirjautuneena olevan käyttäjän id.

## Yiin tapahtumat

Suurin osa Yiin luokista periytyy *CComponent*-luokasta, joka mahdollistaa tapahtumat (*action*). Tapahtuma on viesti siitä, että ohjelman tulee suorittaa joku toiminto. Yhteen käsittelijään pystytään rekisteröimään useita tapausten käsittelijöitä, jotka reagoivat tiettyihin tapauksien tyyppeihin. Käsittelijä voi saada parametreja tapahtumalta ja reagoida niihin. Alla esimerkki käsittelijän ja tapahtuman toiminnasta:

<http://example.com/index.php?r=myController/myAction>

Yllä oleva linkki kutsuu aluksi *myController*-luokkaa ja tarkastaa sen jälkeen löytyykö parametreja. Tässä esimerkissä parametrina löytyy *myAction*. Suoritetaan parametri *myAction*, jonka metodi löytyy *myController* luokasta.

Tässä käydään läpi, kuinka ennalta mukautettuja tapahtumia käytetään omassa sovelluksessasi. Kun luodaan tapahtuma *CComponent*-luokan lapsiluokassa, se on hyvä nimetä alkamaan sanalla *'on'*. Esimerkkinä tästä voi käyttää tapahtumaa *onLoad*.

Tyypillisesti tapahtumia käytetään seuraavalla tavalla:

- Luodaan tapahtuma lisäämällä vastaava metodi.
- Liitetään yksi tai useampi tapahtuman käsittelijä.
- Komponentti herättää tapahtuman käyttämällä *CComponent::raiseEvent()*-metodia.
- Kaikki merkityt käsittelijät ovat kutsuttu automaattisesti toimimaan.

Tapahtumankäsittelijä liitetään tapahtumaan, siinä voidaan käyttää *CComponent::attachEventHandler()*-metodia. Se hyväksyy seuraavat kaksi parametria:

- *\$name*: Tapahtuman nimi.
- *\$handler*: Tapahtuman käsittelijä, tulisi käyttää PHP:n vakiota takaisinkutsu käytäntöä.

PHP sisältää muutaman tavan, jolla määrittää takaisinkutsua:

- Käyttää paikallisia metodeita ja välittää sen nimi *string*-muodossa, esim. *'my\_function'*.
- Käyttää staattista luokkametodia. Tämän kanssa välitys käy taulukon kanssa: `array('ClassName', 'staticMethodName')`.
- Käyttää objektin välittävää metodia: `array($object, 'objectMethod')`.
- Luoda ja välittää tuntemattoman metodin käyttämällä metodia *create\_function*:

```
$component->attachEventHandler('onClick',
create_function('$event', 'echo "Click!";'));
```

```
$component->attachEventHandler('onClick', function($event) {
    echo "Click!";
});
```

- PHP5.3 -versiosta lähtien on voinut käyttää tuntemattoman metodin välittämistä ilman *create\_function*-metodia.

```
$component->onClick=$handler;
// tai:
$component->onClick->add($handler);
```

- Koodin voi pitää lyhyempänä käyttämällä komponentin määrittämiä hoitamaan tapahtuman käsittelijöitä:

```
$component->getEventHandlers('onClick')->insertAt(0, $handler);
```

- Tarkempaan tapahtumien käsittelyyn saa listan käsittelijöistä (*CList*) käyttämällä *CComponent::getEventHandlers*-metodia:

```
$component->getEventHandlers('onClick')->add($handler);
```

- Lisätäkseen tapahtuman käsittelijän ensimmäiseksi tapahtumankäsittelylistaan, käytä:

```
$component->detachEventHandler('onClick', $handler);
```

- Poistaaksesi tietyn käsittelijän, voi siihen käyttää *CComponent::detachEventHandler*-metodia:

- Vaihtoehtoisesti voivat tapahtumankäsittelijät listata ja poistaa käsittelijän sitä kautta.

## Tuonti ja tarvelataus

Erillisen koodin tuomiseen ja tarvittaessa sen lataamiseen Yii:stä löytyy hyvä tapa. Luokkien tuomiseen käytetään *Yii::import()*-metodia. Se ei kuitenkaan sisällytä luokkaa heti, vaan vasta siinä vaiheessa kun sitä kutsutaan ensimmäisen kerran. Samaa luokkaa ei myöskään ladata turhaan useampaa kertaa.

Esimerkissä on hakemistossa `/protected/apis/something/` sovellus, joka tuodaan käytettäväksi `/protected/controllers/TestController.php`-tiedostossa.

```
class TestController extends CController
{
    public function actionIndex($value)
    {
        $lyric = 'Nothing was found.';

        // importing a class
        Yii::import('application.apis.something.SomeApp ');

        $finder = new LyricsFinder();

        if(!empty($value))
            $lyric = $finder->getText($value);

        echo $lyric;
    }
}
```

Lauseella `Yii::import('application.apis.something.SomeApp')` tuodaan ohjelmakoodi osaksi muuta kokonaisuutta. Lauseessa `application` tarkoittaa omaa sovellusta, joka on `protected`-hakemisto. `apis.something` käännetään muotoon `apis/something` ja `SomeApp` viittaa luokkaan, joka halutaan tuoda. Jos halutaan tuoda kokonainen hakemisto, käy se seuraavasti: `Yii::import('application.apis.something.*')`. Suositeltavaa on tuoda luokka kerrallaan eikä koko hakemistoa, jos siellä on useampia luokkia, ja jos kaikkia ei ole tarvetta käyttää. Tarvittavat luokat, jotka haluaa tuoda, voi lisätä myös `main.php`-tiedostoon ja ne ladataan tarvittaessa käyttöön:

```
return array(
    // global imports
    'import'=>array(
        'application.models.*',
        'application.components.*',
        'application.apis.someApp.*',
        'application.apis.someApp.includes.*',
        'application.apis.someAppOld.Finder',
    ),
);
```

Muita aliaksia ovat mm:

- `application`            `www_juurihakemisto/protected`.
- `system`                `www_juurihakemisto/framework`.
- `zii`                    `www_juurihakemisto/framework/zii`.
- `webroot`              `www_juurihakemisto/`.
- `ext`                    `www_juurihakemisto/protected/extensions`.

## Poikkeukset

Yleisesti poikkeus tulee siitä, kun komponentti ei pysty käsittelemään erikoistilannetta. Poikkeuksille on Yiissä kaksi osa-aluetta, missä niitä on hyvä käyttää:

1. Poikkeukset yksinkertaistavat prosessia havaitsemalla ja korjaamalla sovelluksen virheet ja erikoiset tilanteet, kuten tietokantayhteyden epäonnistumisen tai API virhe.
2. Poikkeukset mahdollistavat erilaisten http vastauksien generoinnin.

## Komponenttien konfigurointi

Komponentit ovat omia ohjelmiaan, joita voidaan liittää toisiinsa ja luoda näin laajempia kokonaisuuksia.

```
'components'=>array(
    'db'=>array(
        'class'=>'system.db.CDbConnection',
        'connectionString'=>'mysql:host=localhost;dbname=database_name',
        'username'=>'root',
        'password'=>'',
        'charset'=>'utf8',
    ),
);
```

Yii asetustiedosto main.php sijaitsee hakemistossa /protected/config/, joka sisältää mm. käytössä olevat komponentit. Myöhemmin tullaan käsittelemään tietokannan käyttöönottamisen, joka tapahtuu seuraavasti:

Tällä tavalla komponentti on käytettävissä kaikissa ohjelman osissa. Tähän komponenttiin pääsee käsiksi Yii::app()->db. Yii sisältää kahden tyyppisiä komponentteja. Nämä ovat WWW-sovellus (CWebApplication) ja konsoli sovellus (CConsoleApplication). Molemmat komponentit on peritty luokasta CApplication, jotenka ne jakavat keskenään sisältämänsä komponentit.

Alla on listattuna konsoli- ja WWW-sovellus komponentit:

Komponentin nimi	Komponentin käyttämä luokka	Kuvaus
<b>coreMessages</b>	CPhpMessageSource	Kääntää ohjelmistokehyksen viestit
<b>db</b>	CDbConnection	Hoitaa tietokantayhteydet
<b>messages</b>	CPhpMessageSource	Kääntää sovelluksen viestit
<b>errorHandler</b>	CErrorHandler	Hoitaa PHPn vikatilanteet ja poikkeukset
<b>securityHandler</b>	CSecurityManager	Hoitaa tietoturvan
<b>statePersister</b>	CStatePersister	Globaalin tilan pysyviä menetelmiä
<b>format</b>	CFormatter	Tiedon muodon käsittelijä
<b>cache</b>	CFileCache	Välimuistin käyttö

Lisäkomponentit, jotka ovat yhteensopivia vain WWW-sovelluksien kanssa:

Komponentin nimi	Komponentin käyttämä luokka	Kuvaus
<b>session</b>	CHttpSession	Istuntoon liittyvät toiminnot
<b>request</b>	CHttpRequest	Käsittelee \$_SERVER -muuttujaa ja keksejä
<b>urlManager</b>	CUrlManager	URL-reititin, käsittelee linkit
<b>assetManager</b>	CAssetManager	Hoitaa yksityisten omaisuususerän tietoja
<b>themeManager</b>	CThemeManager	Hallinnoi teemoja
<b>authManager</b>	CPhpAuthManager	Rooleihin perustuvat käyttöoikeudet
<b>clientScript</b>	CClientScript	Hoitaa asiakaspuolen komentosarjoja
<b>widgetFactory</b>	CWidgetFactory	Luo vimpaimet ja tukee niiden teemoja

Omia sovelluskomponentteja, jotka laajenevat CComponent-luokasta, pystyy lisäämään helposti lisäämällä uuden määrittelyn ja osoittamaan sen haluttuun luokkaan.

### Pyynnöt (request)

Pyyntöjen kanssa voidaan käyttää PHP:n superglobaaleja muuttujia \$\_SERVER, \$\_GET tai \$\_POST. Yii tarjoaa tähän paremman tavan käsitellä niitä CHttpRequest -luokan avulla. Se ratkaisee epäjohdonmukaisuuksia eri WWW-palvelimien välillä, hallitsee selainkeksijä ja tuo hieman lisäturvaa. Kyseiseen komponenttiin pääsee käsiksi Yii::app()->getRequest() -metodin avulla. Alla lueteltu, mitä eri osia on mahdollista palauttaa paluuarvona. Paluuarvo on vahvennetulla fontilla:

getUrl()	http://www.example.com/ <b>test/index?var=val</b>
getHostInfo()	<b>http://www.example.com</b> /test/index?var=val
getPathInfo()	http://www.example.com/ <b>test/index</b> ?var=val
getRequestUri()	http://www.example.com/ <b>test/index?var=val</b>
getQueryString()	http://www.example.com/test/index? <b>var=val</b>

### Hello World

Luodaan uusi kontrolleri, joka tulostaa sivulle 'Hello World'-tervehdyksen. Kontrollerin luomiseen voidaan käyttää yii-komentorivityökalua tai luoda se manuaalisesti luomalla hakemistot ja tiedostot itse. Komentorivillä navigoidaan WWW-hakemistoon, mistä löytyy projektin hakemisto, johon kontrolleri halutaan luoda. Tässä hakemistossa suoritetaan komento *'yii shell'*. Tämä avaa yii shellin. Listan komennoista saa komennolla *'help'*. Jos haluaa lisätietoa controller-komennon käytöstä, onnistuu se komennolla *'help controller'*. Uusi controller luodaan komennolla *'controller controllerin-nimi parametri'*. Message-controller parametrilla helloWorld luodaan komennolla *'controller message helloWorld'*. Tämä luo uuden controllerin, joka löytyy tämän jälkeen hakemistosta (C:\wamp\www\demo\protected\views\message) ja selaimen kautta siihen pääsee käsiksi kohdasta: <http://localhost/demo/index.php?r=message/helloWorld>

Message on controllerin nimi ja sille annetaan parametriksi helloWorld, joka avaa kyseisen sivun. Sisältöä kyseiseen sivuun pääsee lisäämään tiedostosta, joka löytyy hakemistosta protected\views\message\helloWorld.php. Controllerissa pitäisi tämän jälkeen olla kaksi funktiota, actionHelloWorld() ja actionIndex(). Ensimmäinen metodi muodostaa näkymän toiminnon kanssa, joka on helloWorld, ja toinen ilman toimintaa, joka on oletusnäkyvä.



## Reititetty pyyntö

Navigoitaessa HelloWorld -sivulle osoitteen <http://localhost/demo/index.php?r=message/helloWorld> kautta analysoi Yii osoitteen. Reititin querystring -muuttuja osoittaa, että controllerID on message. Tämä kertoo Yii:lle, että se reitittää pyynnön MessageController-luokalle, joka löytyy hakemistosta protected/controllers/MessageController.php. Yii käy myös lävitse actionID:n, joka on helloWorld, jotenka Message-Controllerissa kutsutaan metodia actionHelloWorld(). Tämä metodi muodostaa näkymän helloWorld.php, joka sijaitsee hakemistossa protected/views/message/helloWorld.php. Kaikki tämä hoituu automaattisesti.

## Dynaamisen sisällön lisääminen

Sisältöä helloWorld -näkömään pystyy lisäämään helloWorld.php-tiedoston kautta.

Jos halutaan käyttää muuttujia, on ne hyvä lisätä kontrolleriin actionHelloWorld()-metodiin. Sinne voi lisätä esimerkiksi dynaamisen muuttujan \$time, johon määritetään nykyinen aika ja päivämäärä. Tätä voi käyttää sen jälkeen myös näkymän puolella. Näkymän puolelle voi asettaa myös muuttujia, mutta parempi tapa on lisätä ne controllerin puolelle ja ohjata ne toimimaan näkymässä. Controllerilta kutsutaan render() -metodia, jolle annetaan parametriksi 'helloWorld' ja taulukko, mihin asetetaan kaikki halutut muuttujat, jotka halutaan käyttöön myös näkymän puolella. Avainarvoksi annetaan nimi, joka toimii muuttujana näkymän puolella ja arvoksi sama nimi, joka on muuttujalla controllerin puolella. Metodia kutsutaan seuraavalla tavalla actionHelloWorld()-metodin sisällä:

```
$this->render(helloWorld, array('time'=>$theTime));
```

Tämän jälkeen muuttujaa \$time voidaan käyttää näkymän puolella, joka löytyy hakemistosta /protected/views/message/helloWorld.php. Muuttujaa voi käyttää suoraan, sitä ei tarvitse erikseen alustaa.

## Sivujen linkittäminen

Luodaan jo olemassa olevaan MessageControlleriin metodi actionGoodbye(). Lisätään actionGoodbye()-metodiin metodin kutsu:

```
$this->render('goodbye');
```

Luodaan hakemistoon /protected/views/message/ tiedosto goodbye.php. Lisätään goodbye.php tiedostoon sisältöä. Tarkistetaan toimiiko: <http://localhost/demo/index.php?r=message/goodbye>. Linkitetään sivut toisiinsa.

Tapa 1: `<a href="/demo/index.php?r=message/goodbye">Gyydbye!</a>`

Tapa 2: Käytä apuna Yiiin CHtml -avustajaa. Lisää helloWorld -näkyään linkki `<?php echo CHtml::link("Goodbye", array('message/goodbye')) ; ?>` ja toiseen näkyään sama

## Yhdistäminen tietokantaan

Monet sivustot käyttävät tietokantoja. Käyttääksesi tietokantaa, pitää ohjelmalle kertoa, kuinka siihen pääsee käsiksi. Tietokantataulujen luomista ei käydä erikseen tässä läpi. Siihen voi käyttää haluamaan tapaa. Kokoonpanoasetuksista saa aktivoitua haluamansa tietokannan toimimaan:

WebRoot/my\_project/protected/config/main.php

Jos halutaan käyttää SQLite:ä:

```
db=>array(connectionString=>sqlite: ),
```

Yllä olevan koodin kanssa yhteys otetaan SQLite-tietokantaan, joka on valmiina sisällytetty ohjelman runkoon, joka on generoitu aiemmin. Tietokanta sisältää oletuksena ainoastaan yhden taulun, tbl\_user.

Yhteyden luominen onnistuu tuetulla DBMS (Database Management System), ja se onnistuu luomalla uusi CDbConnection-luokka ja instanssia se.

```
$connection=new CDbConnection($dsn, $username, $password);
```

CDbConnection laajenee CApplicationComponent-luokasta, mikä mahdollistaa sen määrittämisen komponenttina. Tämän avulla voidaan komponentin kokoonpano.,m.,,,m.m,

Vaihtoehtona voi käyttää myös MySQL tietokantaa. Tämä vaatii main.php-tiedostoon seuraavan rivin koodia, joka mahdollistaa tietokantayhteyden:

```
'db'=>array('connectionString'=>"mysql:host=localhost;dbname=testdrive"
Username="root" Password="1234")
```

Tietokannan luomista ei käydä tässä läpi. Hakemistossa protected/data löytyy MySQL- ja SQLite-tietokannoille valmiit mallitiedostot, jotka voi ajaa suoraan tietokantaan. Tietokannan luomiseen voi käyttää myös Wampserverin mukana tulevaa phpMyAdmin-työkalua. Sen avulla onnistuu tietokannan ja taulun luominen. Kun tietokanta ja taulu on luotu voi sille luoda Giin avulla mallin, joka sisältää mm. tietokantataulun säännöt, suhteet muihin tauluihin ja hakufunktiot.

## CRUD operaatioiden toteuttaminen

GRUD tulee sanoista Create, Read, Update ja Delete. Giin avulla voidaan luoda tietokantataululle valmis CRUD-operaatio. Avaamalla giin sivun <http://localhost/demo/index.php?r=gii> ja valitsemalla vasemmalta sivulta linkin *Crud Generator*. Vaadittavat kentät ovat malliluokka ja kontrollerin id. Tietokantataulu testitaulu löytyy aiemmin luotuna tietokannasta. Tätä käytetään myös tässä CRUD-esimerkissä. Malliluokka on Testitaulu ja kontrollerin id tulee olemaan sama, mutta pienellä etukirjaimella kirjoitettuna. Preview näyttää generoitavat tiedostot ja samalla niitä pystyy esikatsелеmaan. Generate-painikkeesta saa generoitua tiedostot. Testitaulun sisällön saa nyt listattua kohdasta:

<http://localhost/demo/index.php?r=testitaulu/index>

Tauluun on mahdollista lisätä, poistaa sekä muokata ja listata sisältöä. Jos tietokantataulun rakenteeseen tekee myöhemmin muutoksia, täytyy muutokset tehdä myös tietokantataulun malliin, eli koodiin, joka generoitiin aiemmin. Mahdollista on myös luoda uusi malli muokatulle tietokantataulun rakenteelle, mutta hyvällä suunnittelulla vältetään liika säätäminen.

## Gii code generator

Gii on WWW-pohjainen koodigeneraattori, joka on mukana 1.1.2-versiosta lähtien. Se selkeämpi käyttöliittymältään ja on laajennettavissa. Gii-moduuli ei ole oletusarvoisesti käytössä, vaan se pitää erikseen aktivoida, joka onnistuu protected/config/main.php-tiedostossa. Lisättävä koodinpätkä on seuraava:

```
'modules'=>array(
    'gii'=>array(
        'class'=>'system.gii.GiiModule',
        'password'=>'my_password',
    ),
),
```

Tämän jälkeen moduuliin pääsee käsiksi asettamasi salasanan kanssa. Esimerkissä käytetty salasanaa `'my_password'`. Moduulin saa auki kohdassa:

- <http://localhost/demo/index.php?r=gii>



## Welcome to Yii Code Generator!

You may use the following generators to quickly build up your Yii application:

- [Controller Generator](#)
- [Crud Generator](#)
- [Form Generator](#)
- [Model Generator](#)
- [Module Generator](#)

Luodaan uusi malliluokka tietokantataululle käyttäen gii-generaattoria. Vasemmassa laidassa on listattuna giin eri työkaluja. Uuden mallin saa luotua *'Model Generator'*-linkin takaa. Linkistä aukeaa uusi sivu, johon voi asettaa tietokantataululle etuliitteen, tietokantataulun nimi ja malli-luokan. Malli -luokka on tietokantataulun nimi ilman etuliitettä ja koska se tulee olemaan luokka, nimetään se aina isolla etukirjaimella. Etuliitteen voi laittaa helpottamaan tietokantataulujen luokittelua, jos niitä on projektissa useampia. Pakolliset kohdat ovat taululle annettava nimi ja malli-luokka. Kun tarvittavat tiedot ovat asetettu, pääsee preview -napista esikatselamaan generoitavat tiedostot, ja mitä koodia ne sisältävät. Generate-napista tiedostot lisätään osaksi projektia. Generoitu malli löytyy hakemistosta `/protected/models/`. Tämä on kääre luokka aiemmin luotuun tietokantatauluun.

## TDD-kehitys

### Asennus

Tässä osiossa asennetaan PEAR, Selenium Server ja XDebug Windows-ympäristöön. Yiin versiosta 1.1 lähtien on siihen integroituna testaukseen tarkoitettut ohjelmistokehykset PHPUnit ja Selenium Remote Control. Näitä ei ole pakko käyttää, mutta ne ovat valmiiksi mukana. Myös muita testaukseen tarkoitettuja ohjelmistokehyksiä on mahdollista käyttää. Tässä oppaassa käytetään yllä mainittuja tapoja.

### *Asennusohjeet PEAR Windows:*

Lataa Pear kohdasta: <http://pear.php.net/go-pear.phar> [tarkistettu 18.3.2013]

Suorita asennus komentoriviltä PHP:n hakemistossa seuraavalla komennolla:

```
c:\wamp\bin\php\php5.3.13\ komennolla -> php go-pear.phar
```

Komentorivi kannattaa avata järjestelmänvalvojan oikeuksilla. Käynnistä valikosta haetaan komentorivi hakusanalla 'cmd' ja avataan se painamalla samanaikaisesti näppäimiä Ctrl + Shift ja painaa hiiren valintänäppäimellä komentokehotetta. Asennus tekee muutoksia erinäisiin tiedostoihin, ja ilman järjestelmänvalvojan oikeuksia se ei aina ole mahdollista.

Asennuksen jälkeen PHP:n hakemistosta suoritetaan PEAR\_ENV.reg, joka asettaa ympäristömuuttujat oikein. Ne voi asentaa myös manuaalisesti, mutta valmis rekisteritiedosto helpottaa ja nopeuttaa prosessia, eikä virheitä pääse ilmaantumaan. Asennuksen jälkeen PEAR:in toiminnan voi testata avaamalla uuden komentorivin, johon kirjoittamalla 'pear' ja hyväksyä enterillä. Tämän jälkeen ohjelma tulostaa listan käytettävistä komennoista. Jos jokin meni pieleen, voi asennuksen koittaa suorittaa uudelleen.

Tämän jälkeen tehdään php.ini-tiedostoon yksi muutos. Tiedostosta haetaan rivi 'include\_path', ja jos se on kommentoitu puolipisteellä, otetaan puolipiste sen edestä pois. Tähän polkuun laitetaan PEAR-hakemiston polku. Rivin muokkauksen jälkeen näyttää seuraavalta, riippuen asennuksen sijainnista:

```
include_path = ".; c:\wamp\bin\php\php5.3.13\PEAR\"
```

Huomaa polun alussa oleva piste ja puolipiste. Tallenna tiedosto ja muutokset tulevat voimaan vasta, kun Wampserver käynnistetään uudestaan.

Järjestelmässä voi olla useampi php.ini-tiedosto. Funktiolla phpinfo() saa selvitettyä, mikä asetustiedosto on käytössä. Asetustiedoston sijainti löytyy otsikon 'Loaded Configuration File' alta. Vastaavaan funktioon pääsee käsiksi esimerkiksi <http://localhost/?phpinfo=1> -osoitteen kautta. Tähän löytyy linkki Wampserverin pääsivulta, joka on sama kuin <http://localhost>. Includen toiminnan voi testata lisäämällä PEAR-hakemistoon kansion 'Text' ja sijoittaa sinne 'TestFile.php'-tiedoston, joka tulostaa jotain echo-komennolla. Tämän jälkeen lisätään include johonkin kohtaan omaa koodia: `include_once("Text/TestFile.php")`. Jos tiedoston tulostus näkyy sivulla, on include onnistunut.

Onnistuneen asennuksen jälkeen voidaan asentaa PHPUnit. PEAR:iin lisätään seuraavat kohdat:

- pear channel-discover pear.phpunit.de.
- pear channel-discover components.ez.no.
- pear channel-discover pear.symfony-project.com.

- pear install symfony/YAML.
- pear install phpunit/PHPUnit.

Testikokoonpanossa onnistuttiin lisäämään kaikki muut, paitsi `components.ez.no`. Virheilmoituksessa viitattiin SSL:ään ja siihen, että sitä ei olisi otettu käyttöön asetustiedostossa. Selvittelyjen jälkeen piti `php.ini`-tiedostosta käydä aktivoimaan `php_openssl.dll`-liitännäinen poistamalla sen edestä puolipiste. Tämä pitää mahdollisesti tehdä kahteen eri `php.ini` -tiedostoon. Molemmat löytyvät Wampserverin alihakemistoista. Ensimmäinen löytyy hakemistosta `/bin/php/php5.3.13/php.ini` ja toinen `/bin/apache/apache2.2.22/bin/php.ini`. Tiedoston tallennuksen jälkeen pitää Wampserver käynnistää uudelleen, jotta muutokset tulevat voimaan. Myös komentorivi piti uudelleen käynnistää ja avata se taas järjestelmänvalvojan oikeuksilla. Tämän jälkeen onnistui myös `components.ez.no:n` lisääminen listaan. Ongelma oli aiemmin myös `pear.symfony.com`-kohteen kanssa, mutta korjaantui samalla tavalla kuin aiempi tapaus.

PHPUnit käynnistyy komennolla `'phpunit'`.

PEAR asennus: <http://pear.php.net/manual/en/installation.getting.php>

PHPUnit asennus: <http://www.phpunit.de/manual/3.8/en/installation.html#installation.pear>

### *Asennusohjeet Selenium Server Windows:*

Selenium Server -paketin saa ladattua osoitteesta: <http://code.google.com/p/selenium/downloads/list>  
[tarkistettu 18.3.2013]

Lataa Standalone Java Client, joka oli kirjoitushetkellä versio 2.31.0. Pura paketti sopivaan hakemistoon. Komentorivillä navigoi asennushakemistoon ja käynnistä Selenium Server komennolla `"java -jar selenium-server"`. Paketin täytyy olla standalone, jotta se toimii oikein. Käynnistämisen jälkeen Windows-palomuuri voi vaatia oikeuksia yhdistää internettiin. Tämän jälkeen palvelimen voi jättää taustalle pyörimään.

### *Asennusohje XDebug Windows:*

Liitännäinen voi olla jo valmiiksi asennettuna. Se tulee Wampserverin mukana, mutta jos käyttää jotain toista palvelinohjelmistoa, niin silloin sitä ei välttämättä tule mukana ja asennettuna. Sen voi tarkistaa `php.ini`-tiedostosta hakusanalla `'zend_extension'` ja jos kyseinen hakusana löytyy ja polku `.dll`-muotoiseen `xdebug`-nimen sisältävään tiedostoon on se asennettuna. Kannattaa tarkistaa myös muut `php.ini`-tiedostot. Jos viittaus tiedostosta jo löytyy, ei seuraavaa kohtaa tarvitse tehdä.

Lataa <http://xdebug.com/download.php> [tarkistettu 18.3.2013] Kirjoitushetkellä versio 2.2.1.

Siirrä tiedosto esim. wamp/bin/php/php5.3.13/Zend\_ext-hakemistoon ja lisää laajennus myös php.ini-tiedostoihin seuraavasti:

```
[xdebug]
zend_extension=c:/wamp/bin/php/php5.3.13/ext/php_xdebug_version.dll
```

Asennuksen voi tarkistaa phpinfo()-metodista käyttämällä hakusanaa *'xdebug'*.

## TDD-kehitys

Testit sijoitetaan fixtures-, functional- ja unit- hakemistoihin. Report-hakemisto sisältää generoidut testien raportit. Tämä tarvitsee XDebug-liitännäisen. Yksikkötestaus on Yiissä PHP-luokka, joka on laajennettu luokasta CTestCase. Testiluokan tunnistaa sen loppuosasta, johon tulee sana test. Message-luokan testiluokka nimettäisiin näin MessageTest-luokaksi. Nämä testiluokat sijaitsevat protected/tests/unit/-hakemistossa. Testiluokassa on testimetodeja, jotka ovat nimetty test-etuliitteellä. Esimerkkinä testimetodi voisi olla testMetodi. Toiminnallisuustestaus on myös PHP-luokkia. Ne ovat laajennettu luokasta CWebTestCase. Nämä testit löytyvät hakemistosta protected/tests/functional/. Toiminnallisuustestaus tarvitsee Seleniumin asennuksen toimiakseen.

Pienissä projekteissa testaus ei ole välttämätöntä, mutta projektin suurentuessa testauksen tärkeys nousee esiin. Mitä enemmän tulee olemaan eri toiminnallisuuksia ja niiden välisiä toimintoja, sitä enemmän yhden toiminnallisuuden vajaa toiminta vaikuttaa koko muuhun ohjelmaan.

TDD (Test-driven development) -kehitys on ohjelmiston kehitysmenetelmä. Sen pääperiaate on, että kirjoitetaan nopea testi, joka ei mene läpi. Testi suoritetaan ja huomataan, että se ei mene läpi. Lisätään tarvittava määrä koodia, jotta testi saadaan menemään läpi. Suoritetaan testi ja katsotaan, että se menee läpi. Karsitaan koodista kaikki ylimääräinen pois ja ajetaan testi vielä niin, että se menee läpi. Nämä kohdat toistuvat koko kehitysprosessin ajan. Tarkoituksena on kirjoittaa mahdollisimman pieniä kokonaisuuksia ja testata jatkuvasti, jotta virheitä ei pääsisi ohjelmaan missään välissä.

Kun PHPUnit ja Selenium Server on asennettu ja päällä, voidaan testaus aloittaa. Aluksi tarkistetaan protected/tests/WebTestCase.php-tiedostosta define('TEST\_BASE\_URL'), että se viittaa nykyiseen projektiin.

## Lomakkeen tarkistus

Yii sisältää valmiina sisäänrakennettuja lomakkeen tarkistukseen tarkoitettuja toimintoja. Vaihtoehtoja on kaksi. Ensimmäisessä vaihtoehdossa luokan metodia käytetään tarkistukseen ja toisessa tavassa tehdään erillinen luokka, mikä hoitaa tarkistamisen.

Jos cURL-liitännäistä ei ole otettu käyttöön, niin seuraava koodinpätkä ei toimi. Sen saa aktivoitua php.ini-tiedostosta. Se voi löytyä seuraavista hakemistoista wamp/bin/apache/apache2.2.22/bin tai wamp/bin/php/php5.3.13/. Jos ei ole varma kumpi asetustiedosto on käytössä, voi liitännäisen aktivoida molempiin tiedostoihin toimimaan.

1. Luokassa sijaitsevan tarkistusmetodin käyttö. Aluksi toteutetaan lomake malli luomalla protected/models/SiteConfirmation.php:

```
class SiteConfirmation extends CFormModel {
    public $url;

    public function rules()
    {
        return array(
            array('url', 'confirm'),
        );
    }

    public function confirm($attribute, $params)
    {
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $this->url);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        curl_close($ch);
        if(trim($output)!='code here')
            $this->addError('url', 'Please upload file first.');
```

Yllä oleva esimerkki tarkistaa, löytyykö parametrina annetusta URL:ista 'code here' -teksti.

2. Äsken luotua mallia käytetään MessageController-luokassa, joka löytyy protected/controllers/MessageController, seuraavasti:

```
class TestController extends CController
{
    function actionIndex()
    {
        $confirmation = new SiteConfirmation();
        $confirmation->url = 'http://yiiicookbook.org/verify.html';
        if($confirmation->validate())
            echo 'OK';
        else
            echo 'Please upload a file.';
    }
}
```



3. Yllä oleva koodi suorittaa tarkistuksen, kun sitä kutsutaan. Jos kaikki toimii oikein, sivulla näkyy ainoastaan "OK"-teksti.

SiteConfirmation -mallissa määriteltiin \$url-kenttä ja rules()-metodissa määritellään sille yksi tarkastussääntö, joka on confirm()-metodi. PHP:n metodilla CURL haetaan verify.html:n sisältö palvelimelta ja verrataan sitä 'code here'-merkkijonoon. Jos nämä eivät vastaa toisiaan, käytetään addError()-metodia.

Seuraavassa esimerkissä tarkistusmetodi siirretään malliluokasta erilliseen luokaan. Luodaan tiedosto hakemistoon protected/components/FileValidator.php:

```
class RemoteFileValidator extends CValidator
{
    public $content = '';

    protected function validateAttribute($object,$attribute)
    {
        $value=$object->$attribute;
        $ch = curl_init();
        curl_setopt($ch, CURLOPT_URL, $value);
        curl_setopt($ch, CURLOPT_RETURNTRANSFER, 1);
        $output = curl_exec($ch);
        curl_close($ch);

        if(trim($output) != $this->content)
            $this->addError($object,$attribute,'Please upload file first.');
```

RemoteFileValidator-luokan pitää periytyä CValidator-luokasta ja toteuttaa sen abstraktin validateAttribute()-metodin. SiteConfirmation-malliin muutetaan tarkastussääntö seuraavaksi:

```
array('url', 'RemoteFileValidator', 'content' => 'code here'),
```

Metodin nimi pitää olla sama kuin mallin ja tarkistussäännön, tai muuten Yii yrittää etsiä ulkoista tarkisteluokkaa samalla nimellä.

## CAPTCHA

CAPTCHAN avulla voidaan varmistaa, että lomakkeen lähettäjänä on ihminen eikä botti. Tällä saadaan karsittua turhat lähetykset. Alla luodaan aluksi sähköpostille tarkistuskenttä ja lopuksi lisätään tähän kenttään CAPTCHA-varmistus.

1. Luo uusi malli `protected/models/EmailForm.php` seuraavasti:

```
<?php class EmailForm extends CFormModel
{
    public $email;
    function rules()
    {
        return array(array('email', 'email'),);
    }
}
```

2. Luo uusi kontrolleri `protected/controllers/EmailController.php`:

```
class EmailController extends Controller
{
    public function actionIndex()
    {
        $success = false;
        $model = new EmailForm();

        if(!empty($_POST['EmailForm']))
        {
            $model->setAttributes($_POST['EmailForm']);

            if($model->validate())
            {
                $success = true;
                // handle form here
            }
        }

        $this->render('index', array(
            'model' => $model,
            'success' => $success,
        ));
    }
}
```

3. Luo uusi näkymä `protected/views/email/index.php`:

```

<?php if($success):?>
<p>Success!</p>
<?php endif?>

<?php echo CHtml::beginForm()?>
    <p>
        <?php echo CHtml::activeLabel($model, 'email')?>
        <?php echo CHtml::activeTextField($model, 'email')?>
        <?php echo CHtml::error($model, 'email')?>
    </p>
    <p>
        <?php echo CHtml::submitButton()?>
    </p>
<?php echo CHtml::endForm()?>

```

Tässä vaiheessa sähköpostille on luotu kenttä ja lähetä-nappi. Tämän toiminnan pääsee tarkastamaan kohdassa: <http://localhost/demo/index.php?r=email>. Seuraavaksi lisätään CAPTCHA.

4. Aluksi täytyy muuttaa lomakkeen mallia. Siihen täytyy lisätä \$verifyCode, mikä pitää sisällään tarkastuskoodin, ja joka täytyy antaa ennen lomakkeen lähettämistä.

```

class EmailForm extends CFormModel
{
    public $verifyCode;
    public $email;

    function rules()
    {
        return array(
            array('email', 'email'),
            array('verifyCode', 'captcha', 'allowEmpty'=>
                !CCaptcha::checkRequirements()),
        );
    }
}

```

5. Kontrolleriin täytyy lisätä ulkoinen toiminto CAPTCHAa varten:

```

public function actions()
{
    return array(
        'captcha'=>array(
            'class'=>'CCaptchaAction',),
    );
}

```

## 6. Näkymään lisätään CAPTCHAN kuva:

```
<?php if (CCaptcha::checkRequirements() &&
Yii::app()->user->isGuest):?>
<p>
<?php echo CHtml::activeLabelEx($model, 'verifyCode')?>
<?php $this->widget('CCaptcha')?>
</p>
<p>
<?php echo CHtml::activeTextField($model, 'verifyCode')?>
<?php echo CHtml::error($model, 'verifyCode')?>
</p>
<?php endif?>
```

Tämän jälkeen sähköpostilomakkeessa on CAPTCHA varmennus.

## Muuta

### HTML-toiminnot

#### Pudotusvalikko

Pudotuslaatikon voi tehdä näkymän puolella käyttäen HTML-koodia tai käyttämällä apuluokkaa CHtml seuraavasti:

```
echo CHtml::dropDownList($name, $select, $data, $htmlOptions);
```

- \$name = nimi, jota käytään \$\_POST-muuttujassa lähetettäessä lomaketta.
- \$select = valitulle arvolle merkkijono.
- \$data = sisältö.

```
$htmlOptions = array('width'=>'100', 'style'='color:red;');
```

- \$htmlOptions = taulukko attribuuteille.

\$data -taulukko sisältää avain arvopareja ja sen saa luotua mallista seuraavasti:

```
$data = CHtml::listData(Model::model()->findAll(), 'id_item', 'item_descr');
```

listData() -metodi luo mallista seuraavan tapaisen taulukon:

```
array(value1=>caption1, value2=>caption2 ... );
```

```
CHtml::listData(array $models, string $valueField, string $textField, string $groupField='');
```

- \$models = lista nimikkeistä.
- \$valueField = sarake, jolla täytetään 'value' -tagi.
- \$textField = sarake, jolla täytetään 'caption' -tagi.
- \$groupField = sarake, jota käytetään ryhmittelemään nimikkeet.

## Linkkien luonti

Linkkejä luodaan käyttämällä CHtml-apuluokkaa seuraavasti:

```
echo CHtml::link("text", array('controller/action'));  
// or  
echo CHtml::link("text", $this->createUrl('controller/action'));  
  
// and whenever you want to create URL leading to a page on your site,  
use:  
$this->createUrl('controller/action');
```

Linkkien tyyliä voi vaihtaa protected/config/main.php-tiedostosta. Tämän jälkeen linkit päivittyvät vastaamaan asetettua tyyliä ja toimivat samaan tapaan kuin ennen. Jos linkkiin ei aseta käsittelijää, käytetään nykyistä.