



VAASAN AMMATTIKORKEAKOULU  
VASA YRKESHÖGSKOLA  
UNIVERSITY OF APPLIED SCIENCES

Jussi Isosävi

ARTIKKELIENHALLINTASOVELLUKSEN  
DOKUMENTOINTI JA  
DOKUMENTOINTITYÖKALUN  
KÄYTTÖOHJE

Tekniikka ja liikenne  
2012

## **ALKUSANAT**

Tämä opinnäytetyö on tehty Vaasan ammattikorkeakoulun tekniikan ja liikenteen yksikön tietotekniikan koulutusohjelmassa keväällä 2012. Työ liittyy JBB Dantek -yrityksen tilaustyöhön.

Vaasan ammattikorkeakoulun puolesta valvojana toimi yliopettaja Pirjo Prosi ja JBB Dantek -yrityksestä ohjaajana Daniel Tisza. Haluan kiittää heitä ja myös kaikkia muita, jotka ovat olleet mukana tämän opinnäytetyön toteutuksessa.

Vaasassa 25.5.2012

Jussi Isosävi

## VAASAN AMMATTIKORKEAKOULU

## Tietotekniikan koulutusohjelma

**TIIVISTELMÄ**

Tekijä	Jussi Isosävi
Opinnäytetyön nimi	Artikkelienhallintasovelluksen dokumentointi ja dokumentointityökalun käyttöohje
Vuosi	2012
Kieli	suomi
Sivumäärä	64 + 2 liitettä
Ohjaaja	Pirjo Prosi

---

JBB Dantekin ohjelmiston tarkoituksena on yksinkertaisella järjestelmällä rekisteröidä tietoja konferensseja varten. Sen tarkoituksena on saada tilaisuuksien järjestäjille artikkelitietojen lähettäminen joustavaksi. Lähetetyistä tiedoista konferenssin järjestäjä valitsee osan painettavaksi julkaisuun. Lopputyössäni olen dokumentoinut ohjelmiston.

Keskeisenä työssä on PHP-kielisen ohjelman dokumentointi ja dokumentaation generointi Doxygen- ohjelmalla. Ohjelmiston rakenne uudistui työn aikana. Aluksi tiedostot lähetettiin ylläpitäjän sähköpostiin. Uudistetussa järjestelmässä hyödynnetään palvelimen ominaisuuksia ottamalla käyttöön tietokannat. Työhön sisältyy dokumentointiin käytettävän ohjelman suppea käyttöohje.

Opinnäytetyöhön kuului myös palvelimen ja dokumentoitavan ohjelman asentaminen omalle koneelle testikäyttöä varten.

---

Avainsanat	PHP, tietokanta, dokumentointi, palvelin
------------	--



## SISÄLLYS

ALKUSANAT .....	2
TIIVISTELMÄ.....	3
ABSTRACT .....	4
SISÄLLYS .....	5
KUVA- JA TAULUKKOLUETTELO.....	7
LIITELUETTELO.....	10
TERMIT JA LYHENTEET .....	11
1 JOHDANTO.....	12
2 OHJELMISTON DOKUMENTOINTI .....	13
2.1 Dokumentoinnin merkitys.....	13
2.2 Menetelmät ja standardit.....	13
3 DOKUMENTOITAVA SOVELLUS .....	16
4 KÄYTETYT TEKNIIKAT .....	17
4.1 CAPTCHA .....	17
4.2 .PHP .....	18
4.2.1 PHP 4.....	18
4.2.2 PHP 5.....	19
4.3 SQL .....	19
4.4 WAMP .....	21
4.4.1 WampServer 2 -palvelimen asennus .....	21
4.4.2 Ongelmia asennuksen jälkeen.....	25
4.5 XAMPP-palvelin.....	25
4.5.1 XAMPP-palvelimen asennus .....	25
5 SOVELLUKSEN LÄHDEKOODIN DOKUMENTOINTI.....	31
5.1 Vanhan ohjelman kuvaus ja toiminta.....	31
5.2 Päivitetyn sovelluksen kuvaus ja toiminta.....	36
5.3 Tietokantaan lisäys .....	38
5.3.4 Tietokannan listaus.....	44

6 DOXYGEN .....	46
6.1 Työkalun valinnasta.....	46
6.2 Dokumentointityylit.....	46
6.3 Doxygenin ominaisuudet ja edut käyttäjälle .....	47
7 LÄHDEKODIN DOKUMENTOINTI DOXYGENILLA.....	50
7.1 Doxygenin rakenne ja toiminta .....	50
7.2 Doxygenin avainsanat.....	53
7.3 Dokumentoinnin tulos.....	54
8 YHTEENVETO.....	61
LÄHTEET .....	63

## **KUVA- JA TAULUKKOLUETTELO**

<b>Kuva 1.</b> IEEE:n standardeja	s. 14
<b>Kuva 2.</b> Esimerkki CAPTCHA-tunnisteesta /20/	s. 18
<b>Kuva 3.</b> Erilaisten tietokantapalvelimien yleisyys /14/	s. 20
<b>Kuva 4.</b> Asennuksen aloitussivu	s. 21
<b>Kuva 5.</b> Asennushakemiston valitseminen	s. 22
<b>Kuva 6.</b> Palvelin ja lähtevän postin osoite	s. 23
<b>Kuva 7.</b> Windowsin palomuurin turvatarkastus	s. 23
<b>Kuva 8.</b> Palvelimen pääkäyttöliittymä	s. 24
<b>Kuva 9.</b> WAMP-palvelimen aloitussivu	s. 24
<b>Kuva 10.</b> UAC:n valinnan poisto	s. 26
<b>Kuva 11.</b> Huomautus	s. 26
<b>Kuva 12.</b> Alkusuvi	s. 27
<b>Kuva 13.</b> Asennuksen valintavaihtoehdot	s. 28
<b>Kuva 14.</b> XAMPP on valmis käytettäväksi	s. 29
<b>Kuva 15.</b> Kysytään ohjelman käynnistystä	s. 29
<b>Kuva 16.</b> XAMPP-palvelimen ohjauspaneeli	s. 30
<b>Kuva 17.</b> Vanhan ohjelman toiminta	s. 31
<b>Kuva 18.</b> Artikkelienhallinnan aloitussivu	s. 32
<b>Kuva 19.</b> Artikkelienhallinnan aloitussivun koodia	s. 32
<b>Kuva 20.</b> Artikkelienhallinnan aloitussivu ongelmatilanteessa	s. 33
<b>Kuva 21.</b> Lomake on täytetty oikein	s. 34
<b>Kuva 22.</b> Tiedon lähettäminen on onnistunut	s. 34
<b>Kuva 23.</b> Vanhan ohjelman kaavio	s. 35

<b>Kuva 24.</b> Ilmoittautumistiedot sisältävä sähköposti	s. 36
<b>Kuva 25.</b> Kuittaus ilmoittajan tiedoista järjestäjälle	s. 36
<b>Kuva 26.</b> Päivitetyn sovelluksen toiminta	s. 37
<b>Kuva 27.</b> Tietojen siirtyminen tietokantaan	s. 38
<b>Kuva 28.</b> Ilmoittautuneet-taulun rakenne	s. 39
<b>Kuva 29.</b> Lomakkeen kenttien määrittäminen	s. 40
<b>Kuva 30.</b> Syötekentän tiedot tekstinä	s. 41
<b>Kuva 31.</b> Syötekentän tiedot SQL-komentoa varten	s. 42
<b>Kuva 32.</b> Lomakkeen tiedot tekstinä	s. 42
<b>Kuva 33.</b> Muodostetaan tiedoista SQL-komento	s. 43
<b>Kuva 34.</b> SQL-komento suoritetaan	s. 44
<b>Kuva 35.</b> Tietojen haku tietokannasta lajiteltuna	s. 45
<b>Kuva 36.</b> Järjestäjälle näkyvä osallistujalista	s. 45
<b>Kuva 37.</b> Doxygen-kommentti /6/	s. 47
<b>Kuva 38.</b> Kommentointimerkintä /9/	s. 47
<b>Kuva 39.</b> Kaavio Doxygen-sovelluksen muunnosprosesseista /8/	s. 50
<b>Kuva 40.</b> Tiedostotunnisteet muunnosprosessissa	s. 52
<b>Kuva 41.</b> Dokumentaation aloitussivu	s. 54
<b>Kuva 42.</b> Esimerkinäkymä kommentteista	s. 55
<b>Kuva 43.</b> Dokumentaatiosta haku avainsanalla	s. 55
<b>Kuva 44.</b> Funktion dokumentointi lähdekoodissa	s. 56
<b>Kuva 45.</b> Doxygenin generoima HTML-sivu funktiosta	s. 56
<b>Kuva 46.</b> DOT-työkalun käyttöönotto Doxygenille	s. 57
<b>Kuva 47.</b> Valitaan kaikki kuvaajat	s. 58



<b>Kuva 48.</b> Doxygenin asetukset kaavion tekoa varten	s. 59
<b>Kuva 49.</b> Generoitu luokkakaavio	s. 60
<b>Kuva 50.</b> Generoitu periytymiskaavio	s. 60
<b>Taulukko 1.</b> Ohjelmistoalan Standardointijärjestöjä /3/	s. 15
<b>Taulukko 2.</b> Doxygenilla generoitujen ohjelmien esimerkkejä /7/	s. 46
<b>Taulukko 3.</b> Doxygenin kaikki avainsanat /6/	s. 53
<b>Taulukko 4.</b> Työssä käytetyt avainsanat /6/	s. 54

## **LIITELUETTELO**

**LIITE 1.** Dokumentointityökalun käyttöohje

**LIITE 2.** Index.php

## TERMIT JA LYHENTEET

<b>Lyhenne/Termi</b>	<b>Kuvaus</b>
HTML	Hypertext Markup Language, merkkaukieli, jota käytetään web-sivuissa
Doxygen	Dokumentointiin suunniteltu ohjelmisto, tuottaa lähdekoodiin sijoitetuista kommentteista dokumentaatiota
Tägi	Avainsana, joka kuvaa tiedon sisältöä
Header	Otsikkotieto, joka lisätään tiedoston alkuun
SQL	Structured Query Language, palvelinten kyselykieli, jolla tehdään hakuja relaatiotietokantoihin. Se on riippumaton käytetyistä verkkoprotokollista.
MySQL	Avoimen koodin tietokantapalvelin, joka on luotettava, helppokäyttöinen ja tukee monia käyttöalustoja. Se käyttää SQL-kieltä.

## 1 JOHDANTO

Tämän opinnäytetyön tavoitteena on dokumentoida ohjelmisto ja luoda dokumentointityökalun käyttöohje.

JBB Dantek -yritys sai työn Vaasan yliopistolta. Tehtäväksi tuli suunnitella ja laatia ohjelma artikkelienhallintaan. Tilaustyön tarkoituksena on helpottaa artikkelitiedostojen lähettämistä konferenssien järjestäjille. Lähetetyistä artikkeleista valitaan osa painettavaksi konferenssijulkaisuun.

Tämän ohjelman dokumentoinnista sain aiheen lopputyölleni JBB Dantekin omistajalta. Yrityksellä oli tarvetta ohjelmiston dokumentointiin, mikä sopi mielestäni hyvin suunnitelmiini lopputyötä varten. Dokumentointia on harjoiteltu opiskeluaikana. Nyt voi tehdä työtä, joka liittyy käytännön tarpeeseen.

Alkuvaiheessa toimeksiantajan olosuhteissa tapahtui muutos. Yrityksen saama toimeksianto peruuntui yliopiston järjestelyiden vuoksi. Lopputyön jatkaminen oli kuitenkin mahdollista ja yrityksen aineisto oli käytettävissä dokumentoinnin toteuttamisessa.

## 2 OHJELMISTON DOKUMENTOINTI

Dokumentoinnilla tarkoitetaan ohjelmistojen, lähdekoodin ja toimintojen kuvaamista ja kartoitusta yrityksen tulevan toiminnan kannalta.

### 2.1 Dokumentoinnin merkitys

Ohjelmistoprojektin aikana syntyy normaalisti paljon tietoa, joka pitää saada dokumentaatiomuotoon. Näin saadaan sovelluksen lähdekoodi sekä ohjelmisto pidetyksi ajan tasalla ohjelmiston kehittyessä ja monimutkaistuessa. Ongelmatilanteissa on hyötyä siitä, että ohjelmistosta on ajanmukainen päivitys. /3/

Jos yrityksessä ei tehdä dokumentaatiota riittävän tarkasti tai jätetään jopa tekemättä, koodin ymmärtäminen ja sen korjaaminen on vaikeaa, koodausvirheet voivat lisääntyä ja sen luotettavuus huononee. Riskitekijänä on tällöin koko projektin hallinnan menettäminen. /3/ Paikkansa pitää tässäkin vanha sanonta: vain historiansa tuntevalla on tulevaisuus.

Dokumentointi on selkeästi toteutettu kokonaisuus, jossa tuotetut dokumentit on tehty valitun alan standardin mukaisesti. Tällä tavalla saadaan maksimoitu hyöty.

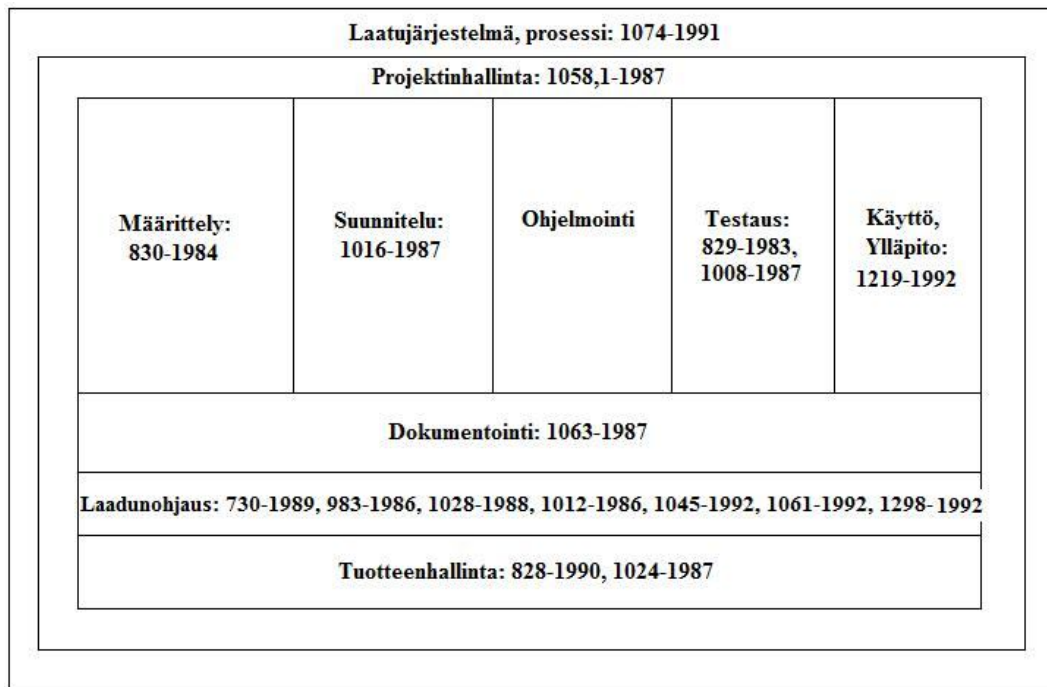
Dokumentoinnin laajuus riippuu aina projektin koosta. Pienessä dokumentaatiossa voidaan osia integroida ja näin pienentää kokonaisuutta. Laajoissa projekteissa pitää tehdä kaikki standardien mukaiset dokumentaation vaiheet. /3/

Dokumentaation ylläpitoa voidaan helpottaa ohjelmien versioinnilla ja luomalla koodia alan mukaisella standardilla alusta alkaen.

### 2.2 Menetelmät ja standardit

Jokaiselle ammattialalle on luotu laaduntarkkailuun liittyviä standardeja (Kuva 1.). Näin saadaan varmistetuksi tuotteiden kelpoisuus ja jatkokehitysmahdollisuudet. Ohjelmointialalla IEEE (sähkö- ja elektroniikkainsinöörien järjestö) tuottaa

alaansa liittyvät standardit. Ne sisältävät säännökset tuotantoprosesseihin ja -vaiheisiin, jotka auttavat luomaan toimivia ohjelmia ja laitteita. /13/



**Kuva 1. IEEE:n standardeja. /3/**

Standardeja voidaan käyttää laatujärjestelmän suunnittelussa ja tässä erityisesti dokumentointimalleja tehtäessä /3/.

Prosessistandardit pyrkivät määrittelemään ohjelmistoprosessia ja asettamaan vaatimuksia koko prosessille tai sen tietyille osille (tuotteenhallinta, testaus ja verifiointi). /3/

Dokumentointimalleja on osina monissa standardeissa. Tämän lisäksi IEEE:n standardikokoelmassa on dokumentointistandardeja, jotka kattavat muun muassa määrittelyn, suunnittelun ja projektisuunnitelman. /3/

**Taulukko 1.** Ohjelmistoalan standardointijärjestöjä /3/.

De Facto	Käytäntö, joka on muuttunut suosituksiksi yleistyessä.
Ulkomaalaiset	
IEEE	Institute of Electric and Electronics Engineers
ANSI	American National Standards Institute
NBS	National Bureau of Standards
NASA	National Aeronautics and Space Administration
DoD	Department of Defence
ISO	International Organization of Standardization
ESA	European Space Agency
Suomalaiset	
KOTEL	Komponenttiteollisuuden yhteistyöjärjestö
SFS	Suomen Standardisoimisliitto
TIEKE	Tietoyhteiskunnan kehittämiskeskus ry.

### 3 DOKUMENTOITAVA SOVELLUS

Tarkoituksena oli tehdä yksinkertainen järjestelmä, jossa Internetin kautta voidaan rekisteröidä tietoja konferenssia varten. Lähetetyistä artikkeleista voi valita osan julkaisuun. Tällä hetkellä ei ole organisaatioita, jotka käyttäisivät tätä tiettyä ohjelmistoa, koska se on kehitysasteella oleva sovellus. Jos löytyy kiinnostuneita yrityksiä, se voidaan muokata niiden tarvitsemiin toimintaympäristöihin.

Kun tiedot lähetetään sähköpostilla, tämä ei vaadi ohjelmistoa pyörittävältä palvelimelta korkeaa suorituskykyä. Tiedot päätettiin pakata zip-tiedostoon, koska se ei aiheuta ongelmaa roskapostisuodattimissa. Alun perin ei haluttu tietokantaa mukaan, koska se vaatii lisää ylläpitoa. Sen lisäksi henkilökuntaa olisi pitänyt kouluttaa. Näistä syistä johtuen tiedot päätettiin lähettää sähköpostilla.

Ohjelmisto sijoitettiin yhteen tiedostoon, jotta se on helppo asentaa kopioimalla palvelimella haluttuun kansioon. Ohjelmistosta jätettiin pois osallistumismaksun maksaminen. Maksuominaisuus olisi vaatinut rahallista panostusta.

Ulkoasua ei ole ohjelmistossa viimeistelty, koska projekti ei edennyt käytännön toteutukseen. Ohjelmistossa on mukana tunnistekuva, jotta voidaan estää automaattisen haittatiedon lähetys. Ohjelmisto toteutettiin PHP-kielellä, koska se on ilmainen ja erittäin laajasti tuettu erilaisilla palvelimilla. Tällä kielellä voitiin myös piirtää tunnistekuva suoraan koodin kautta.

Oleellinen muutos oli tietokantapalvelimen käyttöönotto. Vanhassa versiossa ei varsinaisesti tallennettu tietoja, ne ainoastaan lähetettiin järjestäjän sähköpostiin. Uudessa versiossa syötetyt tiedot tallentuvat tietokantapalvelimelle. Sieltä järjestäjä voi hakea tietoja.



## 4 KÄYTETYT TEKNIIKAT

Tässä luvussa käsitellään sovelluksen toiminnassa käytettyjä tekniikoita.

### 4.1 CAPTCHA

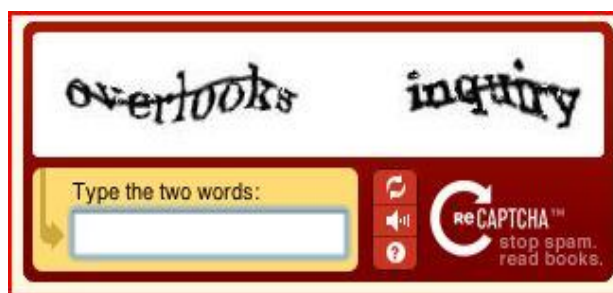
1990-luvun lopulla alkoi muodostua turvallisuusongelmia verkkoympäristön kasvussa. Vuonna 1997 Yahoo! -yrityksen kehittämä Alta Vista -selaimen (verkkohakualgoritmi) käyttäminen ei ollut enää turvallisella tasolla /15/. Tietokoneohjelma keräsi Yahoo Chat Room -palvelusta yksityisiä tietoja, kuten sähköpostiosoitteita markkinointia varten. Roskapostiyritykset alkoivat kehittää ohjelmiaan automaattisiksi. Se tarkoitti sitä, että ne pystyivät luomaan sähköpostitilejä palveluihin muiden tietämättä ja aiheuttivat näin vahinkoa kohteelle. Tämä havahdutti monet eri alojen toimijat tulevasta ongelmasta. Rikollisella toiminnalla aiheutettiin se, että palvelu ei osannut enää erottaa toisistaan ihmisen ja koneen tuotosta. /19/ Oli aika parantaa tietoturvaa.

Vuonna 2000 tietojenkäsittelytieteen tutkijat loivat Carnegie Mellon yliopistossa CAPTCHA-varmenteen /19/. Lyhenne muodostuu sanoista *completely automated public turing test to tell computers and humans apart* /20/. Varmenne on suunniteltu estämään monenlaisia haittatekijöitä, jotka uhkaavat käyttäjien toimintaa sähköisessä tietoliikenteessä.

Yleisesti käytössä on neljä erilaista todennustyyppiä, jotka perustuvat tekstiin, grafiikkaan ja ääneen käyttöön. Rakenne perustuu neljään ominaisuuteen: viivoihin, taustakuviin, tekstiin ja äänen käyttöön. Ne järjestetään satunnaiseen järjestykseen tunnistekuvaan, jotta tietokone ei pysty lukemaan todennuskuvaa.

Tosin nykyään on tullut mukaan monenlaisia muita tapoja, joilla on mahdollista korvata näitä peruselementtejä, kuten kuvien tunnistaminen. Näkörajoitteiset henkilöt on otettu huomioon näiden suunnittelussa edellä mainituissa tekniikoissa. He voivat kirjautua palveluihin ääneen perustuvan tunnisteen avulla.

Yleisin toimenpide on hyödyntää palvelimen hakemistoon tallennettuja tiedostoja eli kuvia, kuvioita ja ääniä. Niiden toiminnallinen funktio on kiinni täysin ohjelman tekijästä, joka on määritellyt koodissa niiden ulkonäön ja toiminnan. Tavallisesti käytetään PHP-ohjelmointikielellä luotua luokkaa, mikä hakee palvelimen hakemistosta vaihtoehtoja käyttäjälle. Luokan sisällä oleva suodattimen metodi arpoa satunnaisesti tunnisteruutuun tekstiä. Tällä tavoin on mahdollista luoda runsaasti erilaisia kuvien ja tekstien yhdistelmiä. /20/



**Kuva 2.** Esimerkki CAPTCHA-tunnisteesta. /20/

CAPTCHA-tunniste on hyvä keino estää väärinkäytöksiä ja konenäön (OCR) tekemiä ongelmia. Varmennetta käytetään nykyään yleisesti sähköisessä tietoliikenteessä. Sitä tarvitaan suojaamaan tärkeitä osia. Se on suunniteltu estämään automaattista tilirekisteröimistä, keskustelualueitten roskapostia (spam), salasanojen koneellista hakemista ja kirjautumista sekä suojaamaan salasanojen palauttamista. /20/

## 4.2 PHP

Alkujaan lähdekoodissa ohjelmointikielen versiona oli PHP4. Lopputyössä käytetty versio on PHP5.

### 4.2.1 PHP 4

PHP (*Hypertext Preprocessor*) on HTML-koodiin upotettava skriptikieli, joka on suunniteltu palvelimille /16/. Rasmus Lerdorf loi kielen 1994, alun perin se luotiin

kävijälaskuriksi. Ohjelmointikieli on tarkoitettu palvelin pohjaisten dynaamisten WWW-sivujen luomiseen. Tuki eri alustoille ja käyttöjärjestelmille on kattava ja koodia voidaan luoda monilla ohjelmistoilla. /11/

Andi Gutmans ja Zeev Suraski jatkoivat PHP:n kehitystyötä 2000-luvun alussa. Ohjelmistoon tehtiin uusi ydin, Zend Engine. Tarkoitus oli tehostaa sovelluksen toimintaa ja PHP-koodin modulaarisuutta. Ytimen nimi muodostuu kehittäjien nimistä. Se paransi muun muassa kielen rakennetta ja tukea moniin web-palvelimiin sekä lisäsi käyttäjien turvallisuutta. Oliio-ohjelmointi tuli PHP 3:ssa mukaan. Aikaisemmissa versioissa ei ollut sitä ominaisuutta saatavana. PHP 5:ssä vasta tuli laaja tuki. Opinnäytetyön ohjelmakoodi on luotu PHP 4 -versiolle. /17/

#### **4.2.2 PHP 5**

Tässä uudessa päivityksessä optimoitiin Zend Engine 2.0 -ydintä. Siinä muokattiin oliomallia ja kymmeniä uusia ominaisuuksia. /17/

Se on heikosti tyypitetty kieli. Koodissa ei tarvitse määritellä tietotyyppisiä, kuten int /16/. Ohjelman ajon aikana voi tulla odottamattomia tuloksia, jos yritetään laskea epäyhteensopivilla tietotyypeillä (miinus-lasku kirjaimella ja numerolla). Muistinhallinta on automaattinen. /17/

#### **4.3 SQL**

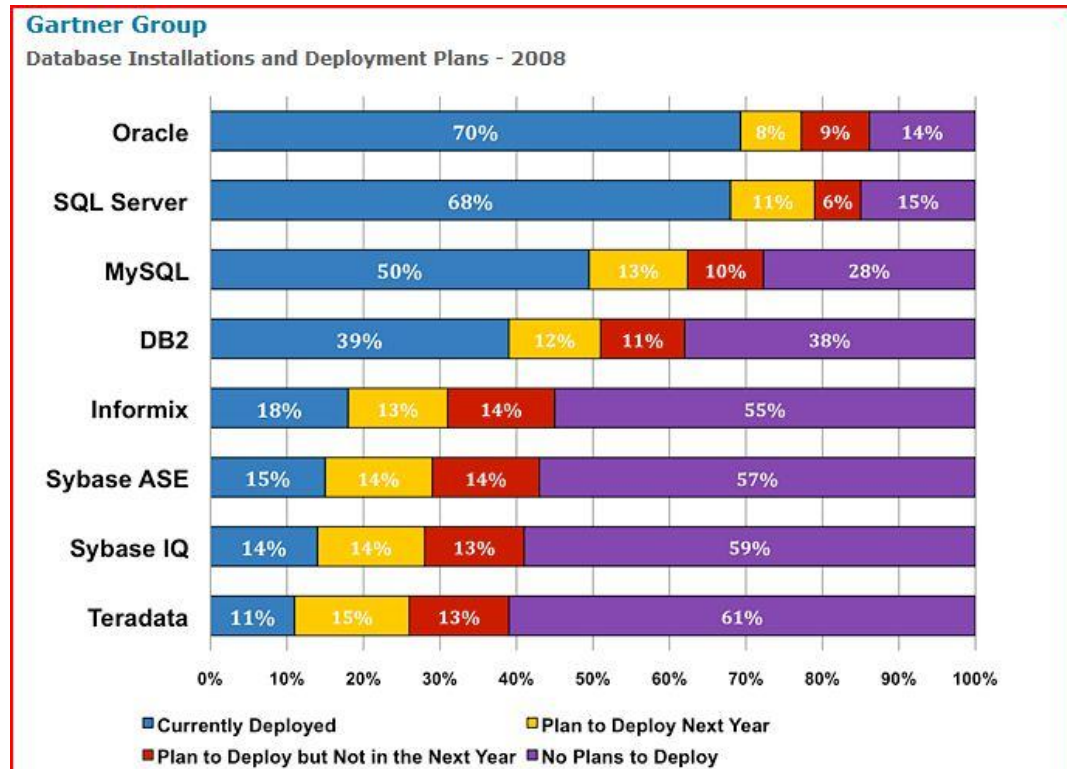
IBM kehitti 1970-luvulla tietokantakielen Structured English Query Language (SEQUAL), joka luotiin alun perin tietokannalle DB2. Väliversio oli nimeltään SEQUAL/2. SQL-kieli (Structured Query Language) on kaikkien nykyaikaisten tietokannanhallintajärjestelmien taustalla. 1980-luvun alussa Amerikan kansallinen standardointijärjestö (ANSI) tunnusti relaatiotietokantoja hyödyntävän kielen erinomaisuuden. Tämän jälkeen alkoivat yleistyä relaatiotietokantojen palvelimet, esimerkiksi Oracle ja MySQL. /21/

SQL on relaatiotietokantakieli. Sitä käytetään tietokantojen kyselyihin, päivityksiin ja ylläpitoon. Tietokantaan tallennettuja tietoja voidaan sen avulla hakea, poimia ja suodattaa. /12/ Yksi tärkeimmistä eduista on, että SQL-kieli on riippumaton käytössä olevasta ohjelmistokielestä, editointiohjelmasta ja käyttöliittymästä /10/.

### MySQL-palvelin

MySQL-palvelin on yksi SQL-kieltä käyttävä tietokantapalvelin. Sen ovat ottaneet käyttöön suuret yritykset: Google, Facebook ja Adobe. MySQL tukee yli 20 alustaa, esimerkiksi Linux, Windows, Mac OS, Solaris, IBM ja AIX. Nykyään yleinen tietokantaohjelmisto on ilmainen MySQL Community Edition. /14/

Alla oleva kuva osoittaa erilaisten tietokantojen käyttöosuudet. Työssäni käytin MySQL:ää, joka sijoittuu neljän parhaan joukkoon. /14/



**Kuva 3.** Erilaisten tietokantapalvelimien yleisyys. /14/

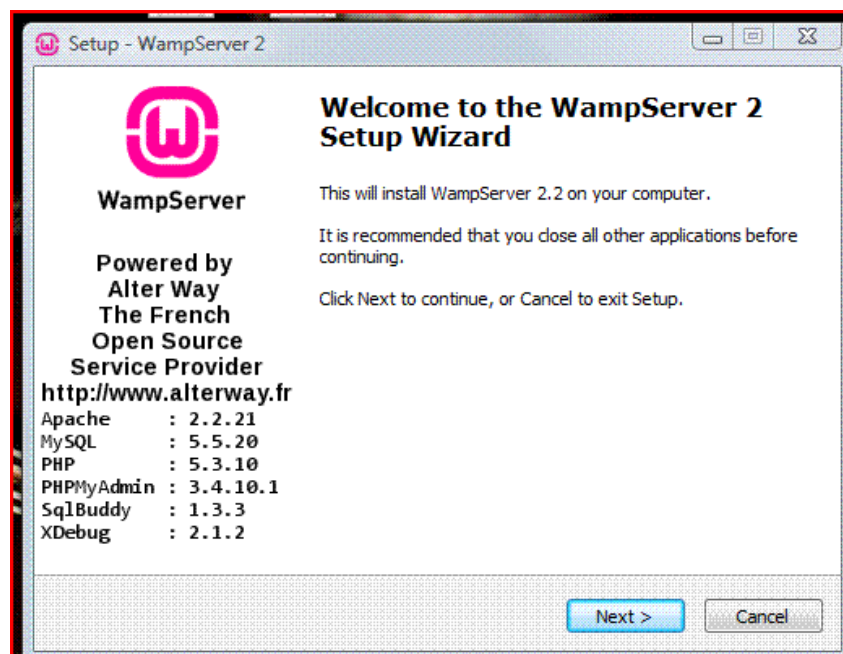
## 4.4 WAMP

Ohjelmistokokonaisuus sisältää monia palveluita: Apache 2.2.21, Php 5.3.10, Mysql 5.5.20, XDebug 2.1.2, XDC 1.5, PhpMyadmin 3.4.10.1, SQLBuddy 1.3.3 ja webGrind 1.0.

Wamp on palvelinohjelmisto-paketti, jonka asentamalla voi omalla koneella kehittää ja testata web-sovelluksia. Se on helposti hallittava ja muokattava ohjelmisto. Siitä on tarjolla versiot monelle käyttöjärjestelmälustalle. Tässä projektissa käytetään vain Windowsin 32 bittistä ohjelmistoa (WAMP). Uusimpia ohjelmistopaketteja on tarjolla myös 64 bittisinä.

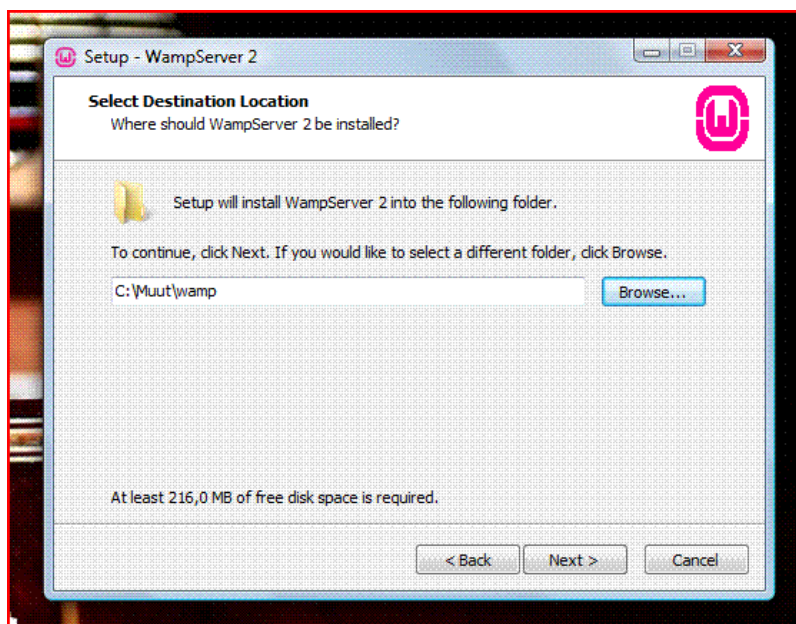
### 4.4.1 WampServer 2 -palvelimen asennus

WampServer 2.2D 32 bit -versio ladattiin SourceForge -sivustolta (<http://www.wampserver.com/en/>) ja asennettiin koneelle. Operaatio kesti vain muutaman minuutin. Ohjelman asennus oli yksinkertaista seuraavien vaiheiden kautta.

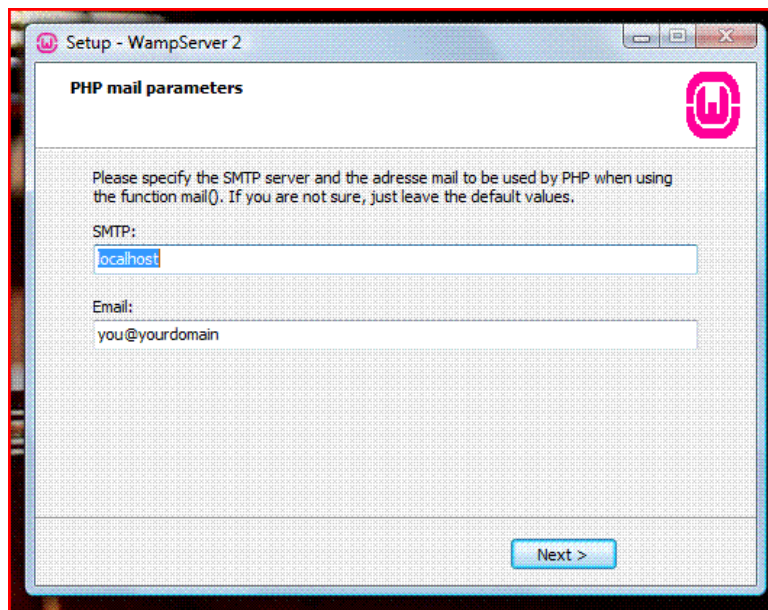


**Kuva 4.** Asennuksen aloitussivu.

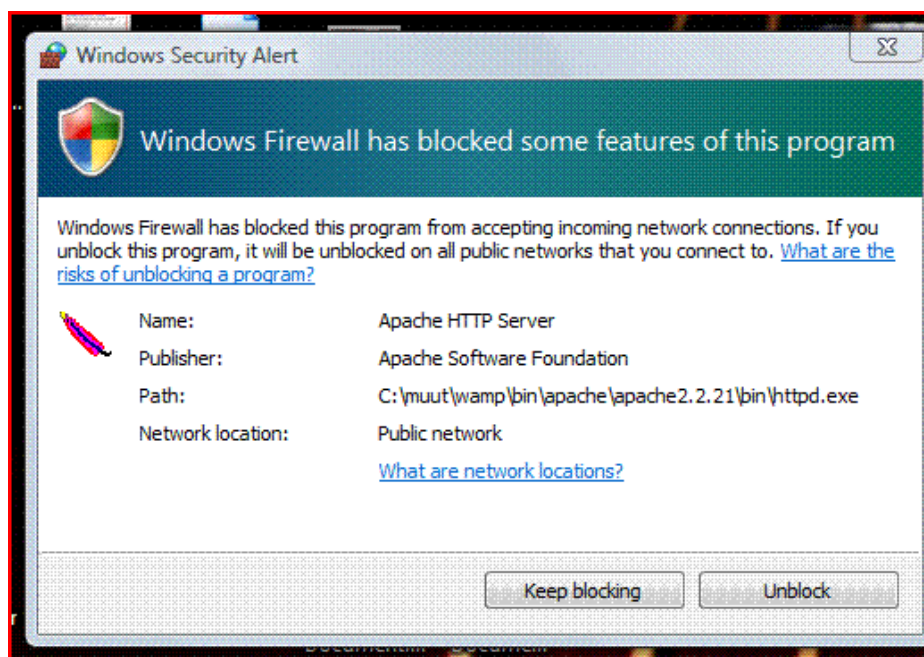
Valitaan aluksi ohjelman asennushakemisto (Kuva 5.), sitten määritellään palvelin ja lähtevän postin osoite (Kuva 6.). Tämän jälkeen Windows kysyy palomuurin asetuksia, hyväksytään ne (Kuva 7.). Asennetaan ohjelma ja käynnistetään sitten palvelin ohjelman pääkäyttöliittymästä (Kuva 8.). Sen kautta päästään WAMP-palvelimen aloitussivun (selain) näkymään, josta voidaan muokata myös tietokantapalvelimen asetuksia (Kuva 9.).



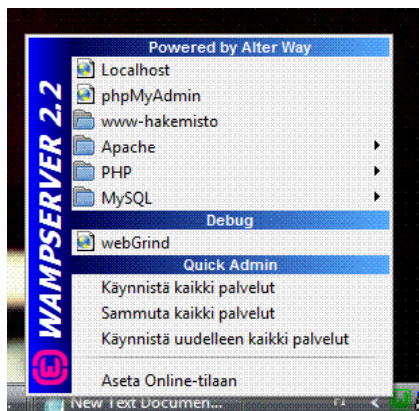
**Kuva 5.** Asennushakemiston valitseminen.



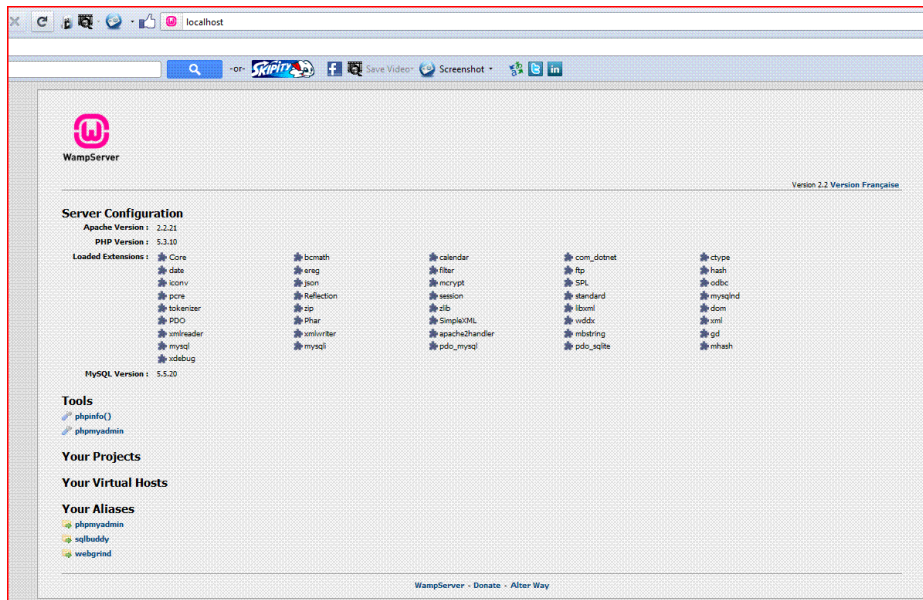
**Kuva 6.** Palvelin ja lähtevän postin osoite.



**Kuva 7.** Windowsin palomuurin turvatarkastus.



**Kuva 8.** Palvelimen pääkäyttöliittymä.



**Kuva 9.** WAMP-palvelimen aloitussivu.

Palvelimen asennus on valmis. Seuraava askel on luoda tietokanta käyttämällä ohjelmapaketissa olevaa phpMyAdmin -palvelua. Se on sovellus, jolla voi hallita palvelinten asetuksia ja luoda tietokantoja. Tarkoituksena on hyödyntää tietokantoja tiedon tallentamisessa.



#### 4.4.2 Ongelmia asennuksen jälkeen

Asennuksen jälkeen ilmeni ongelmia ohjelman oikeuksien kanssa. Jostakin syystä phpMyAdmin-hallintasivu ei auennut normaalisti. Manuaalisesti PHPMyAdmin -salasanan ohittaminen oli työlästä. Hallintasivun lopulta auettua ei silti päästy MySQL -palvelimelle. Sen root-salasanan ohittamisen jälkeen päästiin luomaan tietokantaa, mikä oli alun perin tavoitteena. Ratkaisu ei ollut toimiva, koska ongelma toistui jokaisen käynnistyksen jälkeen. Tämä johti uuden palvelimen valintaan.

#### 4.5 XAMPP-palvelin

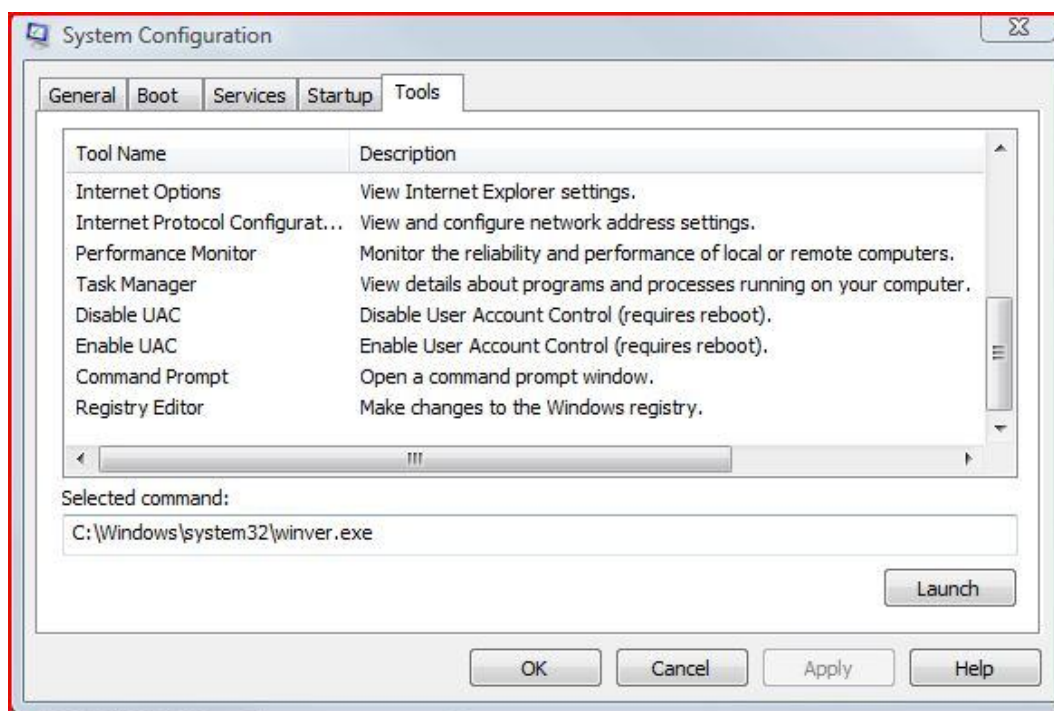
XAMPP (Cross-plattform Apache http Server MySQL PHP Perl) on monen suunnittelijan tuote, jonka tarkoituksena oli luoda ilmainen ja helppokäyttöinen palvelinohjelmisto. Se kehitettiin vuonna 2002. Alkujaan tarkoituksena oli luoda sovellus, joka auttaa ohjelmistokehittäjiä toimintojen testauksessa. Se on julkaistu GPL-lisenssin alla. Yleisenä palvelimena käytettynä se ei ole turvallisin valinta. /22/

##### 4.5.1 XAMPP-palvelimen asennus

Käyttöoikeuksien tiukkuuden takia piti poistaa WampServer 2 -palvelin. Se vaihdettiin XAMPP-palvelimeen. Sen voi ladata osoitteesta <http://www.apachefriends.org/en/xampp-windows.html>.

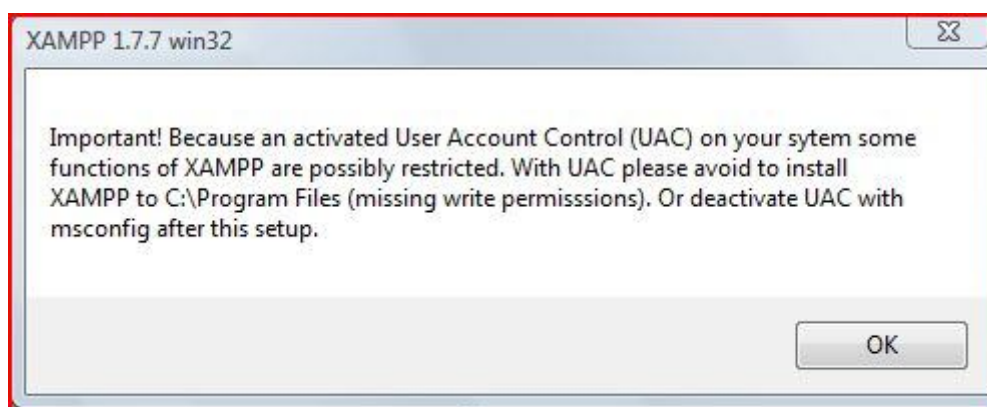
Asennuksen alussa on huomioitava muutama tärkeä asia. Microsoft Windows Vista -versiossa käyttötilikontrolli (UAC) voi vaikuttaa ohjelman ominaisuuksien toimintaan. On suositeltavaa asentaa ohjelma muualle kuin Windowsin ohjelmakansioon, jotta saadaan kirjoitusoikeus tiedostoihin. UAC:n voi ottaa tilapäisesti käytöstä, jolloin XAMPP toimii paremmin.

Käyttötilikontrollin säätäminen voidaan tehdä käyttöjärjestelmän muokkaamiseen tarkoitetulla msconfig-ohjelmalla. Siitä valitaan Tools-välilehti (Kuva 10.) ja sieltä valitaan disable UAC ja painetaan launch-painiketta. Muutoksen jälkeen käynnistetään käyttöjärjestelmä uudestaan, jotta asetukset tulevat voimaan. /18/



**Kuva 10.** UAC:n valinnan poisto.

Seuraava huomautus voi ilmestyä, jos UAC:a ei ole poistettu.

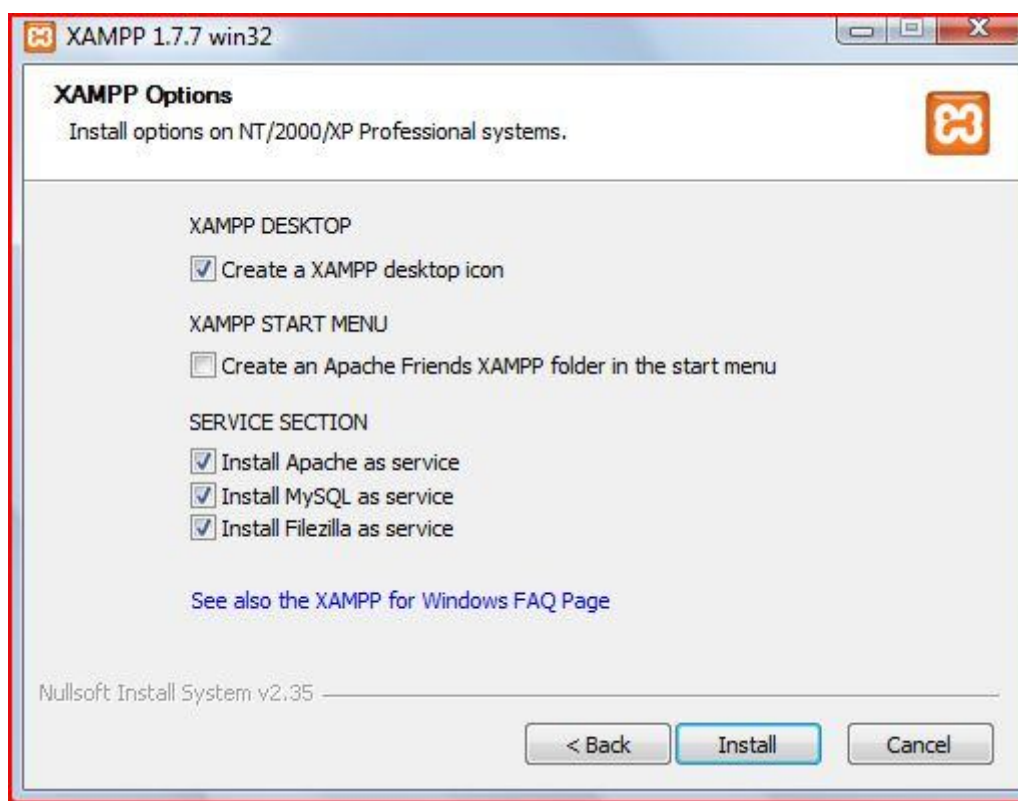


**Kuva 11.** Huomautus.

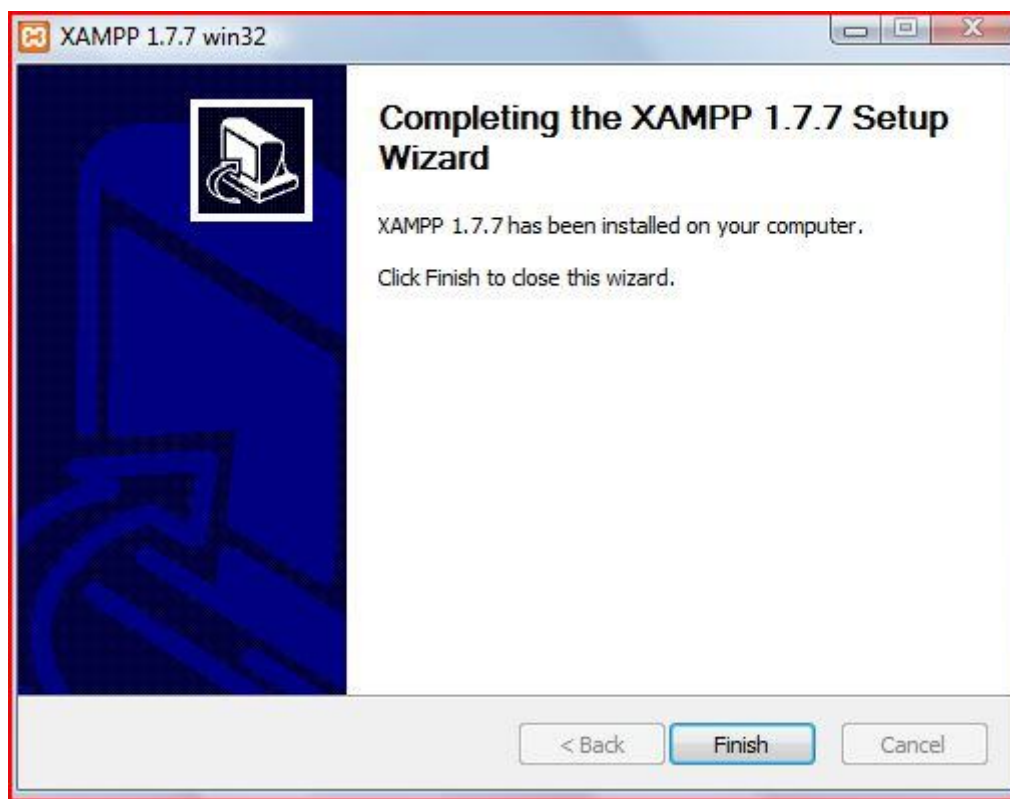
Aloitetaan asennus ohjelman avulla (Kuva 12.). Sitten määritetään haluttu asennushakemisto ja seuraavassa valitaan asennukseen liittyvät optiot (Kuva 13.) ja aloitetaan asennus. Tämän jälkeen asennusavustaja ilmoittaa, että XAMPP on valmis käytettäväksi (Kuva 14.).



**Kuva 12.** Alkusuvi.



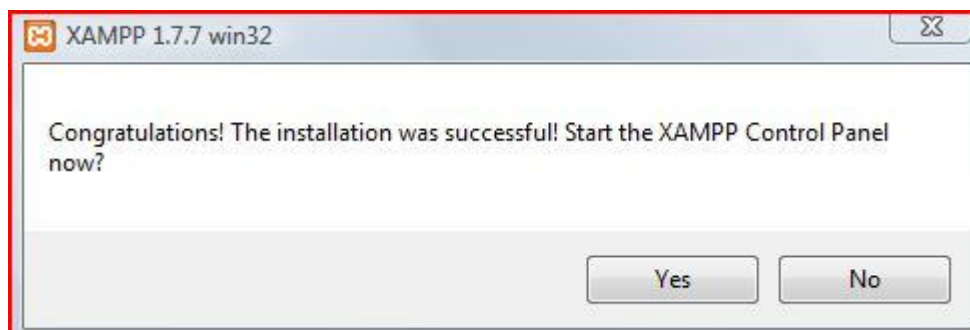
**Kuva 13.** Asennuksen valintavaihtoehdot.



**Kuva 14.** XAMPP on valmis käytettäväksi.

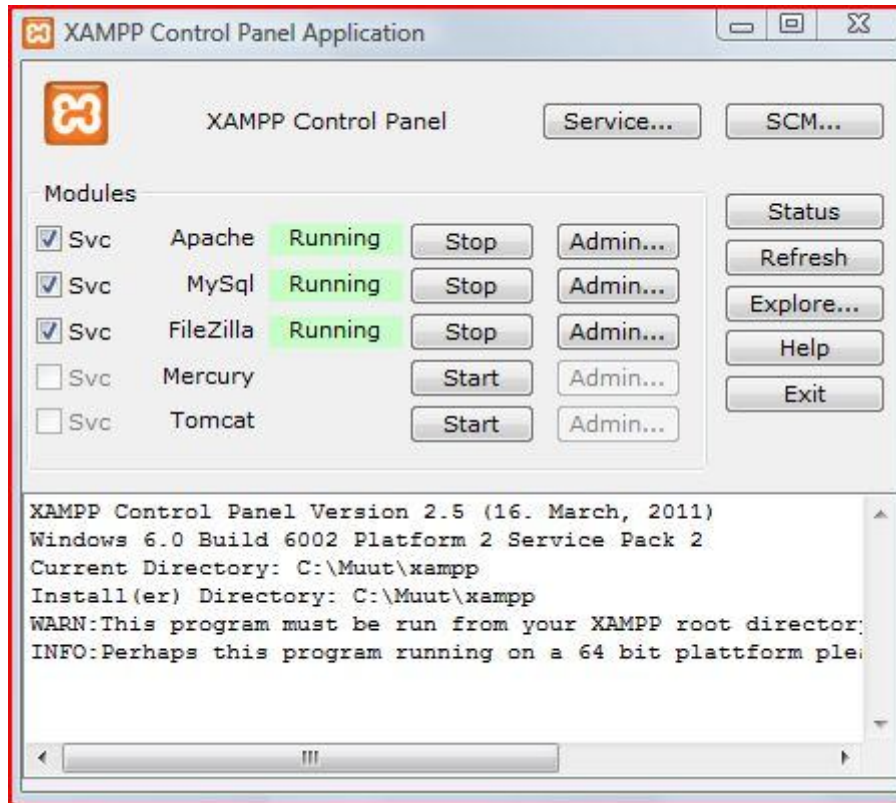
Asennuksen aikana Windowsin palomuri voi kysyä ohjelman oikeuksista liikennöidä palomuurin läpi. Hyväksytään tietoliikenteen läpikulku. On muistettava poistaa edellinen palvelin järjestelmästä tai muuttaa ohjelman liikennöintiporttia.

Asennuksen jälkeen kysytään vielä, halutaanko ohjelma käynnistää heti.



**Kuva 15.** Kysytään ohjelman käynnistystä.

Tästä ohjauspaneelistä valitaan moduulit ja hallintamääritteet.

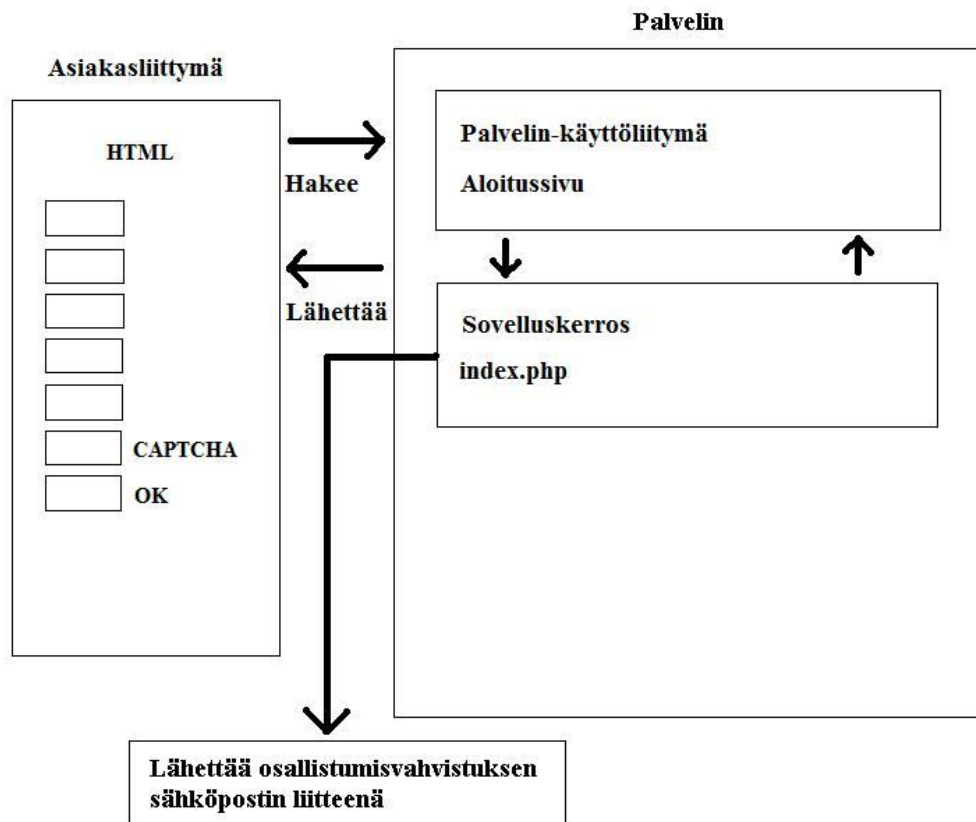


**Kuva 16.** XAMPP-palvelimen ohjauspaneeli.

## 5 SOVELLUKSEN LÄHDEKOODIN DOKUMENTOINTI

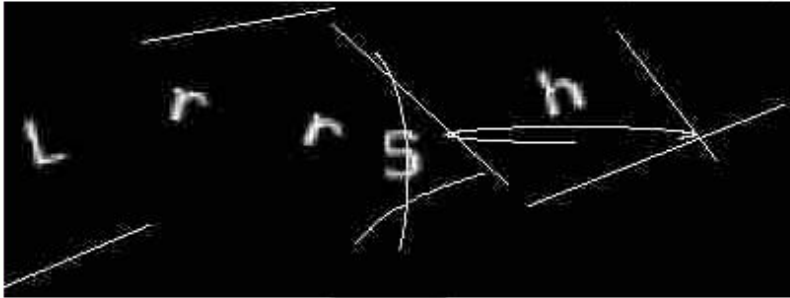
### 5.1 Vanhan ohjelman kuvaus ja toiminta.

Osallistuja täyttää PHP-koodin luoman lomakkeen pyydetyillä tiedoilla. Tämän jälkeen sovelluskerros tarkastaa tiedot. Jos kaikkia pyydettyjä tietoja ei ole annettu, palautetaan vajaa lomake ja pyydetään täydentämään puuttuvat tiedot.



**Kuva 17.** Vanhan ohjelman toiminta.

First Name  Must be non-empty  
 Last Name  Must be non-empty  
 Email Address  Must be non-empty  
 Institution  Must be non-empty  
 Publication Title  Must be non-empty  
 Publication File   Must be non-empty



Verification Code in Image

**Kuva 18.** Artikkelienhallinnan aloitussivu.

```

/**
 * @title Määrittää lomakkeen
 */

$captcha_img_url = $_SERVER['SCRIPT_NAME'].'?captcha=1&nocache='.md5( rand() );

$form = new Form( $_SERVER['SCRIPT_NAME'], 'post',
                 'multipart/form-data', $captcha_img_url );

$form->add_input( new Input( 'First Name', '', '', 20, 'Must be non-empty' ) );
$form->add_input( new Input( 'Last Name', '', '', 20, 'Must be non-empty' ) );
$form->add_input( new Input( 'Email Address', '', '', 20, 'Must be non-empty' ) );
$form->add_input( new Input( 'Institution', '', '', 40, 'Must be non-empty' ) );
$form->add_input( new Input( 'Publication Title', '', '', 40, 'Must be non-empty' ) );
$form->add_input( new Input( 'Publication File', 'file', '', 30, 'Must be non-empty' ) );

```

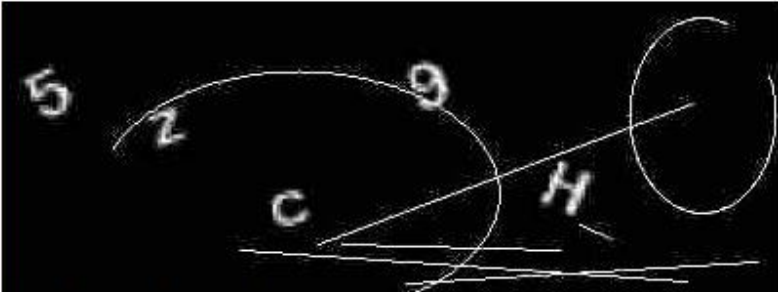
**Kuva 19.** Artikkelienhallinnan aloitussivun koodia.



Käyttäjä kirjoittaa lomakkeeseen halutut tiedot. Tunnisteella varmistetaan, että käyttäjä on ihminen. Käyttäjän hyväksytyt tiedot, ne lähetetään koodissa (index.php) merkittyyn hallinnoijan sähköpostiosoitteeseen liitteenä (zip-pakettina).

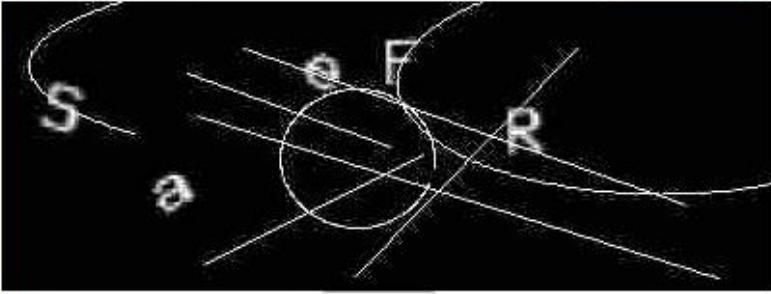
Seuraavilla sivuilla on esimerkkikuvia ohjelman toiminnasta.

Käyttäjä on jättänyt lomakkeeseen tyhjiä syöttökenttiä. Puuttuvasta tiedosta jää ilmoitus "Must be non-empty". Tässä tilanteessa käyttäjä joutuu täyttämään tyhjät syöttökentät palautuvaan lomakkeeseen.

First Name	<input type="text" value="Pekka"/>
Last Name	<input type="text" value="Meiklinen"/>
Email Address	<input type="text"/> Must be non-empty
Institution	<input type="text" value="Meiklinen Oy"/>
Publication Title	<input type="text" value="Meik"/>
Publication File	<input type="text"/> <input type="button" value="Browse..."/> Must be non-empty
	
Verification Code in Image	<input type="text"/>
<input type="button" value="Submit"/>	

**Kuva 20.** Artikkelienhallinnan aloitussivu ongelmatilanteessa.

First Name	<input type="text" value="Pekka"/>
Last Name	<input type="text" value="Meiklinen"/>
Email Address	<input type="text" value="Meika@zen.com"/>
Institution	<input type="text" value="Meiklinen Oy"/>
Publication Title	<input type="text" value="Meik"/>
Publication File	<input type="text" value="C:\Users\Jussi\Desktop\56983"/> <input type="button" value="Browse..."/> Must be non-empty



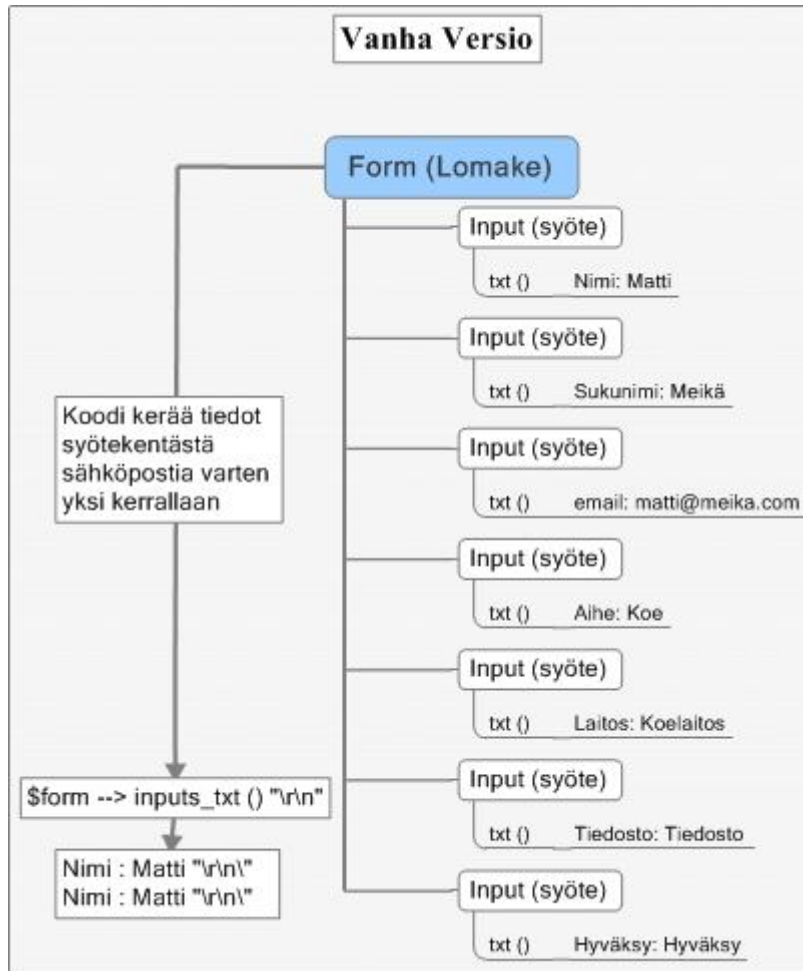
Verification Code in Image

**Kuva 21.** Lomake on täytetty oikein.

Information Submitted.  
File Submitted.

**Kuva 22.** Tiedon lähettäminen on onnistunut.

Kuvasta selviää ohjelman toiminta kooditasolla.



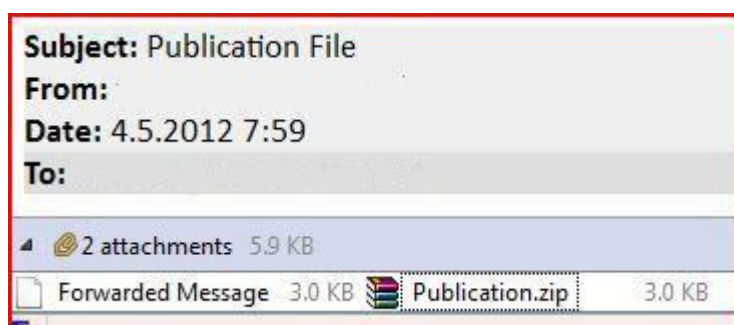
**Kuva 23.** Vanhan ohjelman kaavio.

### Ohjelman lähettämät sähköpostiviestit

Ohjelma lähettää konferenssin järjestäjälle tarkoituksella kaksi eri tiedostoa. Jos käyttäjän antama tiedosto on liian iso tai jää filteriin, ilmoittautumistiedot tulevat varmasti perille omassa viestissään.



**Kuva 24.** Ilmoittautumistiedot sisältävä sähköposti.



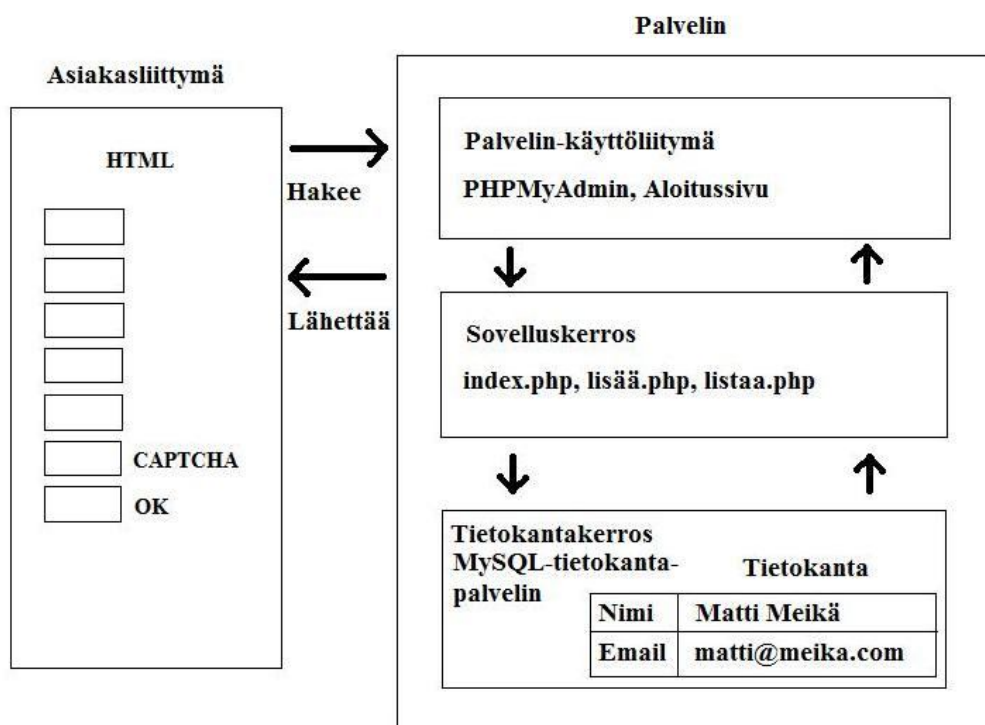
**Kuva 25.** Zip-pakatun artikkelin sisältävä sähköposti.

## 5.2 Päivitetyn sovelluksen kuvaus ja toiminta

Päivitetyn version perustana on tietokantapalvelimen käyttö. Osallistujat kirjoittavat tiedon WWW-lomakkeeseen varatuille paikoille. Asiakasliittymään ei ole tehty muutoksia. Muutoksia on tehty sovellukseen tietokerroksessa ottamalla käyttöön tietokantapalvelin.

Palvelimella tiedon lisäys hoidetaan PHP- ja SQL-ohjelmointikielen avulla (index.php), tietokantaan lisätyjä tietoja voidaan katsoa eri ohjelmalla (listaa.php).

Tarkoituksena on muokata index.php -sivua siten, että käyttäjän kirjoittama tieto siirtyy suoraan MySQL-palvelimelle, jossa sijaitsee tietokanta.



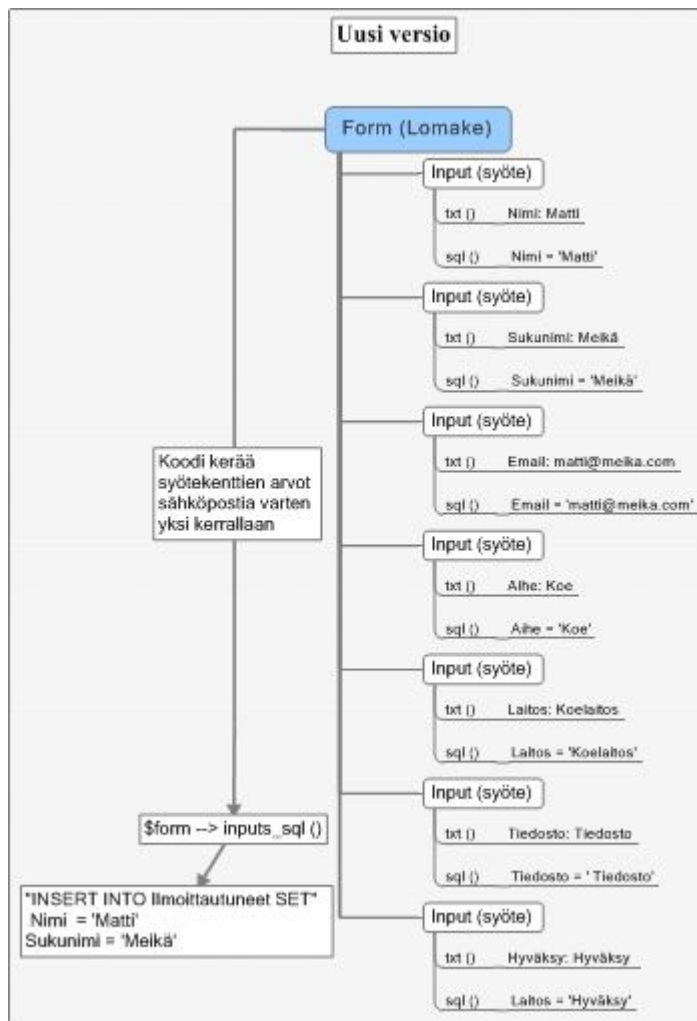
**Kuva 26.** Päivitetyn sovelluksen toiminta.

PHP-koodi muodostaa HTML-koodia, joka lähetetään käyttäjän selaimelle. HTML-koodia ei tallenneta missään vaiheessa tiedostoon, se syntyy aina haku-toiminnon aikana.

#### **Sovellus toimii kolmella tasolla:**

Asiakasliittymä on osallistujalle näkyvä osa. Sovelluskerros pitää yhteyttä tietokantaan ja välittää hyväksymiskuittauksen osallistujan käyttöliittymään. Tietokerros sisältää tiedot ja tietokantapalvelimen. /11/

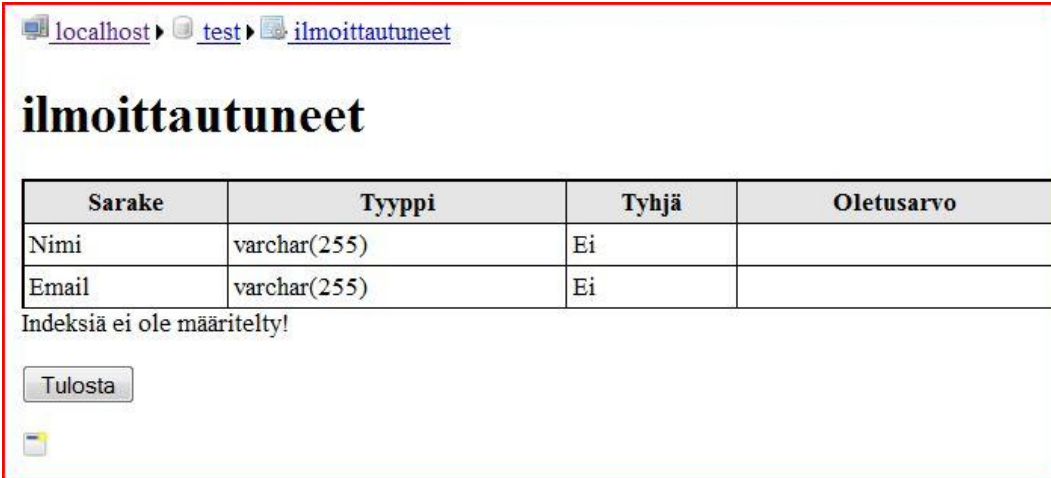
Osallistuja täyttää PHP-koodin luoman lomakkeen pyydetyillä tiedoilla. Tämän jälkeen sovelluskerros lisää tiedot tietokantaan SQL-komennolla. Jos kaikkia pyydettyjä tietoja ei ole annettu, palautetaan vajaa lomake ja pyydetään täydentämään puuttuvat tiedot. Tietokerroksessa oleva ohjelma lisää tiedot tietokantaan. Ohjelma lähettää viestin osallistujalle onnistuneesta ilmoittautumisesta.



**Kuva 27.** Tietojen siirtyminen tietokantaan.

### 5.3 Tietokantaan lisäys

MySQL-palvelimella luotiin test-tietokanta. Sinne tehtiin taulu ilmoittautuneille. Tauluun lisättiin kaksi kenttää *Nimi* ja *Email*. Nille määriteltiin tyyppiä merkkijono.



localhost › test › ilmoittautuneet

## ilmoittautuneet

Sarake	Tyyppi	Tyhjä	Oletusarvo
Nimi	varchar(255)	Ei	
Email	varchar(255)	Ei	

Indeksiä ei ole määritelty!

Tulosta

**Kuva 28.** Ilmoittautuneet-taulun rakenne.

Perusavaimella eli indexillä saadaan jokainen tietokannan taulun rivi erilaiseksi ja MySQL:ssä voi myös valita perusavaimen automaattisesti kasvavan numeroinnin /2/. Tätä ominaisuutta ei ole nyt hyödynnetty, mutta se olisi seuraava kehitysaskel.

Esimerkki SQL-komennosta, jolla lisätään uusi rivi tauluun.

```
INSERT INTO Ilmoittautuneet SET
```

```
    Nimi='Matti',
```

```
    Email='Matti@Matti.com'
```

Komennon alkuosa on aina sama ja jokainen taulun kenttä on komennossa esimerkin mukainen. Komennossa olevat kentät erotetaan pilkulla. Tämä komento-muoto on MySQL:n INSERT-käskyn laajennus. Seuraavaksi tuleva komento on muotoiltu SQL-kielen standardin mukaisesti /13/. Siinä lisätään uusia rivejä ja annettuja arvoja aiemmin luotuun taulukkoon.

*INSERT INTO Ilmoittautuneet*

*(Nimi, Email) VALUES ('Matti', 'Matti@Matti.com')*

Sovelluserroksessa koodia muokataan. Kaksi syötekenttää muutetaan *Nimi*- ja *Email*-nimisiksi, jotta ne vastaavat taulun kenttien nimiä. Muut lomakkeen kentät on otettu pois käytöstä. Työn varsinainen tarkoitus on tutustua sovelluksen dokumentointiin ja toissijaisesti tutustua PHP -kieleen ja MySQL-tietokantaan tulevaisuutta varten.

```
416
417 /*****
418  * Define form
419  *****/
420
421 $captcha_img_url = $_SERVER['SCRIPT_NAME'].'?captcha=1&nocache='.md5( rand() );
422
423 $form = new Form( $_SERVER['SCRIPT_NAME'], 'post',
424                 'multipart/form-data', $captcha_img_url );
425
426 $form->add_input( new Input( 'Nimi', '', '', 20, 'Must be non-empty' ) );
427 // $form->add_input( new Input( 'Sukunimi', '', '', 20, 'Must be non-empty' ) );
428 $form->add_input( new Input( 'Email', '', '', 20, 'Must be non-empty' ) );
429 // $form->add_input( new Input( 'Laitos', '', '', 40, 'Must be non-empty' ) );
430 // $form->add_input( new Input( 'Julkaisun aihe', '', '', 40, 'Must be non-empty' ) );
431 $form->add_input( new Input( 'Publication File', 'file', '', 30, 'Must be non-empty' ) );
432
```

**Kuva 29.** Lomakkeen kenttien määrittäminen.



### Lisäykset Input-luokkaan.

Syötekenttä sisältää txt-funktion, joka palauttaa syötekentän nimen ja käyttäjän syöttämän tiedon kaksoispisteellä erotettuna. Tätä funktiota kutsutaan, kun halutaan tekstimuodossa syötekentän tiedot.

```
54 //Kirjoitetun tiedon lisäys
55 function txt( )
56 {
57     if( strcmp( $this->type, 'submit' ) == 0 ||
58         strcmp( $this->type, 'file' ) == 0 )
59     {
60         return;
61     }
62
63     $txt = $this->name.' : '.$this->value;
64
65     return $txt;
66 }
```

**Kuva 30.** Syötekentän tiedot tekstinä.

SQL-komentoa varten syötekenttä pitää laittaa seuraavaan muotoon:

*Nimi='Matti'*

```
67 //tietokantaan lisäys
68 function sql( )
69 {
70     if( strcmp( $this->type, 'submit' ) == 0 ||
71         strcmp( $this->type, 'file' ) == 0 )
72     {
73         return;
74     }
75
76     $sql = $this->name."='".$this->value."'";
77
78     return $sql;
79 }
80
```

**Kuva 31.** Syötekentän tiedot SQL-komentoa varten.

### Lisäykset Form-luokkaan

Lomake sisältää inputs\_txt-funktion, joka poimii kaikki käyttäjän lomakkeelle syöttämät tiedot rivinvaihdolla erotettuna. Tätä funktiota kutsutaan, kun halutaan tekstimuodossa lomakkeen tiedot.

```
194
195 // Syöttökenttien tietojen kokoaminen lomakkeelta
196 function inputs_txt( )
197 {
198     $txt = '';
199
200     foreach ( $this->inputs as $key => $val )
201     {
202         $txt .= $val->txt( )."\r\n";
203     }
204
205     return $txt;
206 }
207
```

**Kuva 32.** Lomakkeen tiedot tekstinä.

Lomakkeelle lisätiin `inputs_sql`-funktio. Se tekee lomakkeen tiedoista SQL-komennon. Funktiossa käydään läpi kaikki lomakkeen syötekentät ja kutsutaan kentän `sql`-funktia. Se palauttaa syötekentän tiedot sopivassa muodossa SQL-komentoa varten. Tieto liitetään komentoon.

```
206
207 //Lomakkeen tiettojensyöttö-komento tietokantaan
208 function inputs_sql( )
209 {
210     $sql_cmd = 'INSERT INTO Ilmoittautuneet SET ';
211
212     //käydään läpi kaikki lomakkeeseen lisätyt kentät
213     foreach ( $this->inputs as $key => $val )
214     {
215         //hyppää yli jos kenttä on tyhjä
216         if( strcmp($val->sql( ), '' ) != 0 )
217         {
218             $sql_cmd .= $val->sql( ).", ";
219         }
220     }
221
222
223     //poistetaan viimeinen pilkku ja välilyönti
224     $sql_cmd = substr($sql_cmd, 0, strlen($sql_cmd)-2);
225
226     return $sql_cmd;
227 }
```

**Kuva 33.** Muodostetaan tiedoista SQL-komento.

Pääohjelmassa avataan yhteys tietokantaan lomakkeelta saatujen tietojen lisäämiseksi. Tässä vaiheessa kutsutaan `inputs_sql`-funktiota, joka palauttaa lomakkeen tiedot muutettuna SQL-komennoksi. Lopuksi komento suoritetaan MySQL-tietokantapalvelimella.

```
/******  
* Lisätään tietokantaan  
*****/  
  
// avataan yhteys tietokantaan('palvelin', 'käyttäjä', 'salasana')  
$db = mysql_connect('localhost','root','');  
  
// valitaan tietokanta  
mysql_select_db("test",$db); // oma tietokanta  
  
// tehdään SQL-lause  
// lisätään lomakkeen kaikki syöttökentät samannimisinä käyttäjän antamilla arvoilla  
$sql = $form->inputs_sql();  
  
echo $sql;  
  
// ajetaan SQL-lause tietokannassa  
$result = mysql_query($sql,$db);
```

**Kuva 34.** SQL-komento suoritetaan.

### 5.3.4 Tietokannan listaus

SQL-koodi lajittelee osallistujat ensin etunimen, sitten sähköpostiosoitteen mukaan aakkosjärjestykseen.

```
SELECT *FROM Ilmoittautuneet ORDER BY Nimi, Email ASC
```

```

listaa.php
<h1>ILMOITTAUTUNEET </h1>
<hr>
<form name="form1" method="post" action="">
  <table width="286" border="1">

<?php
// testataan yhteys tietokantaan('palvelin', 'käyttäjä', 'salasana')
$db = mysql_connect('localhost','root','');

// valitaan tietokanta
mysql_select_db("test",$db); // oma tietokanta

// tehdään SQL-lause
// valitaan ilmoittautuneet-taulusta kaikki ja järjestetään nimet kasvavasti eli aakkosjärjestykseen
$sql = "SELECT * FROM Ilmoittautuneet ORDER BY Nimi ASC";

// ajetaan SQL-lause tietokannassa
$result = mysql_query($sql,$db);

//tulostettan haetut rivit taulusta
while($row = mysql_fetch_array($result))
{
    print "<tr>";
    print "<td>";
    print $row['Nimi'];
    print "</td>";

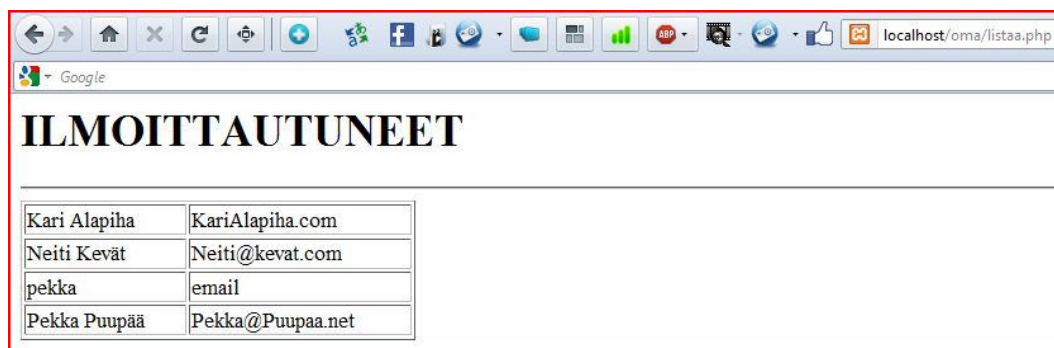
    print "<td>";
    print $row['Email'];
    print "</td>";
    print "</tr>";
}
?>

</table>
</form>
<br>

```

**Kuva 35.** Tietojen haku tietokannasta lajiteltuna.

Tiedot sijaitsevat tietokannassa tallennettuna aina samassa järjestyksessä. Haettaessa ne voidaan lajitella mihin järjestykseen tahansa. Lajittelu koskee ainoastaan yksittäistä hakua.



**Kuva 36.** Järjestäjälle näkyvä osallistujalista.

## 6 DOXYGEN

### 6.1 Työkalun valinnasta

JBB Dantek -yritys tutustui eri dokumentointisovelluksiin ja arvio niiden soveltuvuutta vaatimuksiin ja tarpeisiin. Valitun ohjelman vaatimuksena on tukea monenlaisia käyttöjärjestelmiä ja ohjelmistoalustoja. Se päättyi valitsemaan Doxygen-ohjelman laajasta sovellusvalikoimasta.

Doxygen on ohjelma, joka käsittelee lähdekoodin ja siihen lisätyt kommentit sekä generoi sen dokumentiksi. Dimitri van Heesch loi ohjelman C++-kielellä ja lisensoi Doxygen-ohjelman vuonna 1997. Monet suuret yritykset ovat ottaneet ohjelman käyttöönsä avoimen koodin projekteissa [5]. Markkinoilla on tähän tarkoitukseen monia muitakin samankaltaisia sovelluksia: ForgeDoc ja Doc-O-Matic.

Seuraavassa taulukossa on muutama esimerkki Doxygenilla generoiduista projekteista:

**Taulukko 2.** Doxygenilla generoitujen ohjelmien esimerkkejä. [7]

264/AVC -kodeekki	<a href="http://iphone.hhi.de/suehring/tml/doc/index.htm">http://iphone.hhi.de/suehring/tml/doc/index.htm</a>	( viitattu 20.3.2012)
Flac -audiokodekki	<a href="http://flac.sourceforge.net/api/">http://flac.sourceforge.net/api/</a>	( viitattu 20.3.2012)
OpenMSX Emulator	<a href="http://openmsx.sourceforge.net/docs.php">http://openmsx.sourceforge.net/docs.php</a>	( viitattu 20.3.2012)
Navit-autonavigaattori	<a href="http://www.navit-project.org">http://www.navit-project.org</a>	( viitattu 20.3.2012)

### 6.2 Dokumentointityylit

Doxygenilla kommentoidaan suoraan lähdekoodiin ohjelman toiminnan osia (funktio/luokka). Vaikka koodin eri osiin liittyy erilaisia tietoja, voidaan Doxygenilla kuitenkin käyttää dokumentointiin samanlaisia lohkoja. Jokaisessa lohkossa on eri määrä avainsanoja. Jokaiseen avainsanaan lisätään selitys. Seuraavaksi on kaksi esimerkkiä kommentointityyleistä.

Koodin kommentti voidaan aloittaa kahdella tavalla eli ” /\*! ” tai ” /\*\* ”.

```

/*!
 * \brief      Pretty nice class.
 * \details    This class is used to demonstrate a number of section commands.
 * \author     John Doe
 * \author     Jan Doe
 * \version    4.1a
 * \date       1990-2011
 * \pre        First initialize the system.
 * \bug        Not all memory is freed when deleting an object of this class.
 * \warning    Improper use can crash your application
 * \copyright  GNU Public License.
 */
class SomeNiceClass {};

```

**Kuva 37.** Doxygen-kommentti. /6/

Koodia voidaan merkitä kahdella tavalla ”\brief” tai ”@brief”.

```

/**
 * @title Tunnisteen generointi
 * @return arvottu tunnisteteksti
 */

```

**Kuva 38.** Kommentointimerkintä. /9/

### 6.3 Doxygenin ominaisuudet ja edut käyttäjälle

Doxygen toimii monissa ohjelmistoalustoissa, kuten Windows, Linux ja Mac OSX. Se tukee useita ohjelmointikieliä (Java, C, C++, C Sharp, Cobra, VHDL, PHP, IDL, Fortran jne.). /5/

Ohjelma generoi automaattisesti luokkia, kaavioita HTML-muotoon ja sallii viittauksia dokumentaatioon (dokumentteihin) muista Doxygen-projekteista tai muihin projektin osiin. Sillä voi kääntää monia kolmannen osapuolen apuformaatteja, esimerkiksi HTML:ää. Dokumentaatio voidaan siirtää uudelleen generoimatta toiseen paikkaan. Doxygen tukee monia merkkikoodauksia ja käyttää UTF-8:a sisäisesti ja ulostulogenerointiin. Doxygen tunnistaa automaattisesti yleiset, suojatut ja yksityiset ohjelmaosiot. Jos koodi on kommentoitu oikealla tyyllillä, ohjelma tunnistaa tägit automaattisesti. /4/

Lähdekoodin osat on muoto-opillisesti korostettu helppolukuisiksi. Joustava kommenttien sijoittelu sallii dokumentaation header-tiedoissa, lähde-tiedoissa tai erillisissä tiedostoissa. /4/

Dokumenttiin voi kirjoittaa normaaleja HTML-tägeja. Tämän lisäksi dokumentaation voi generoida automaattisesti LATEX-tiedostoksi, rtf-tiedostoksi tai MAN-ohjesivuksi. /4/

Eri käyttöjärjestelmille on saatavilla monia ohjelmia projektien tarpeiden ja laajuuden mukaan. Jokainen sovellus tukee erilaisen määrän ohjelmointialustoja, ohjelmointikieliä ja ominaisuuksia. /4/

Jokaisella ohjelmalla on omanlaisensa ohjelmistojen käyttöoikeudet eli lisenssit. Niitä on monenlaisia ja monen tasoisia eri käyttöjärjestelmille. Käyttäjän pitää määritellä projektinsa pohjalta, tarvitseeko maksullista ja suljettua lisenssiä, joka estää vapaan käytön enemmän kuin avoin lisenssi. Avoimessa käyttöoikeudessa on myös omat rajoituksensa, jotka vaikuttavat toimintatapoihin.

Doxygen-ohjelmistoa jaetaan GPL-käyttöoikeuden (GNU General Public License) alaisuudessa /5/. Se on vapaiden ohjelmistojen lisenssi. Sitä voi kuka tahansa kopioida, jakaa ja muokata. Tärkein lisenssiin liittyvä perusominaisuus on, ettei ohjelman virheiden korjaamiseen ja uusien tarpeellisten ominaisuuksien lisäämiseen tarvitse kysyä lupaa. Kukaan ei omista koodia. Muokatuissakin GPL-koodiin perustuvissa ohjelmistoissa oikeudet ja vapaudet pysyvät ennallaan.

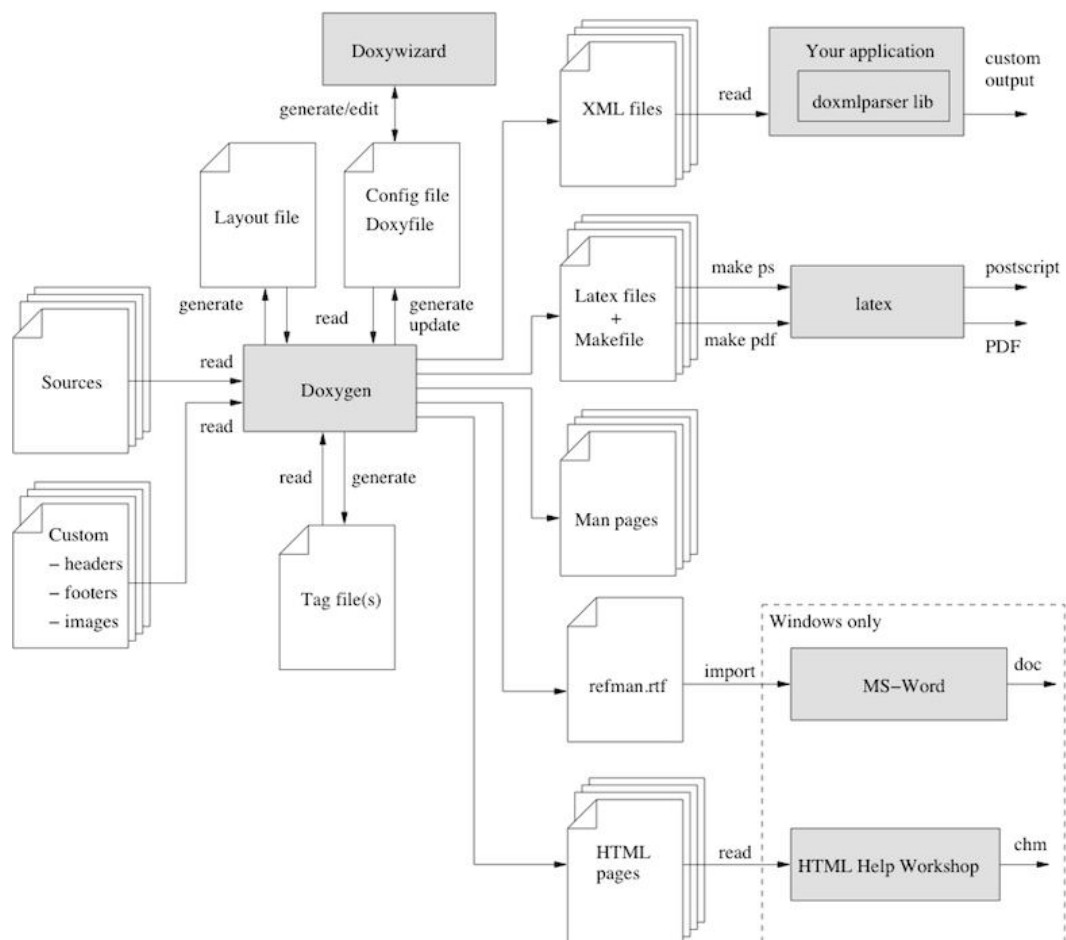
Yksi tärkeimmistä Doxygenin valintaan johtavista syistä oli laaja käyttöjärjestelmien tuki. Sitä voidaan käyttää unix-tyylisissä systeemeissä Mac SX - ja Windows-järjestelmissä.



Ohjelma tukee seuraavia ohjelmointikieliä: C++, C, Java, Objective-C, Python, IDL (Corba ja Microsoft-tyyliset), Fortran, VHDL, PHP ja C#. Muutama näistä on nykyään vähäisessä käytössä /5/. Edellä olevassa erityisen tärkeitä on soveltuvuus PHP-kieleen.

## 7 LÄHDEKOODIN DOKUMENTOINTI DOXYGENILLA

### 7.1 Doxygenin rakenne ja toiminta



**Kuva 39.** Kaavio Doxygen-sovelluksen muunnosprosesseista. /8/

Aluksi Doxygen hakee konfiguraatiotiedostosta tiedot suorittamista varten. Tämä tiedosto voidaan luoda tai sitä voidaan muokata Doxywizardin avulla.

Konfiguraatitiedostossa on määritetty lähdekooditiedostot tai lähdekoodihakemisto. Doxygen käy läpi lähdekooditiedostot ja tuottaa dokumentaation halutuissa tiedostomuodoissa. Doxygen ei itse generoi kaavioita, mutta pystyy automaattisesti käyttämään ulkoista apuohjelmaa kaavioiden tekemiseen ja upottamaan ne dokumentaatioon. /8/

Komentoriviltä voi käyttää konfiguraatitiedostoa seuraavalla komennolla: *doxygen -g <config-file>./8/*

Kuvassa on lueteltu erilaisia lähdekoodipäätteitä, jotka ohjelma tunnistaa automaattisesti.

<b>Tunniste</b>	<b>Kieli</b>
.IDL	IDL
.DDL	IDL
.odl	IDL
.Java	Java
.CS	C#
.d	D
.php	PHP
.php4	PHP
.php5	PHP
.Inc	PHP
.phtml	PHP
.m	Objective-C
.M	Objective-C
.mm	Objective-C
.py	Python
.f	Fortran
.sillä	Fortran
.f90	Fortran
.vhd	VHDL
.VHDL	VHDL
.Tcl	TCL
.ucf	VHDL
.qsf	VHDL
.MD	Markdown
.markdown	Markdown

**Kuva 40.** Tiedostotunnisteet muunnosprosessissa.

Doxygenin ja Doxywizardin käyttöohje on liitteessä 1.

## 7.2 Doxygenin avainsanat

**Taulukko 3.** Doxygenin kaikki avainsanat. /6/

<code>\a</code>	<code>\e</code>	<code>\if</code>	<code>\private</code>	<code>\subsubsection</code>
<code>\addindex</code>	<code>\else</code>	<code>\ifnot</code>	<code>\privatesection</code>	<code>\tableofcontents</code>
<code>\addtogroup</code>	<code>\elseif</code>	<code>\image</code>	<code>\property</code>	<code>\test</code>
<code>\anchor</code>	<code>\em</code>	<code>\implements</code>	<code>\protected</code>	<code>\throw</code>
<code>\arg</code>	<code>\endcode</code>	<code>\include</code>	<code>\protectedsection</code>	<code>\throws</code>
<code>\attention</code>	<code>\endcond</code>	<code>\includelinen</code>	<code>\protocol</code>	<code>\todo</code>
<code>\author</code>	<code>\enddot</code>	<code>\ingroup</code>	<code>\public</code>	<code>\tparam</code>
<code>\authors</code>	<code>\endhtmlonly</code>	<code>\internal</code>	<code>\publicsection</code>	<code>\typedef</code>
<code>\b</code>	<code>\endif</code>	<code>\invariant</code>	<code>\ref</code>	<code>\union</code>
<code>\brief</code>	<code>\endinternal</code>	<code>\interface</code>	<code>\related</code>	<code>\until</code>
<code>\bug</code>	<code>\endlatexonly</code>	<code>\latexonly</code>	<code>\relates</code>	<code>\var</code>
<code>\c</code>	<code>\endlink</code>	<code>\li</code>	<code>\relatedalso</code>	<code>\verbatim</code>
<code>\callgraph</code>	<code>\endmanonly</code>	<code>\line</code>	<code>\relatesalso</code>	<code>\verbinclude</code>
<code>\callergraph</code>	<code>\endmsc</code>	<code>\link</code>	<code>\remark</code>	<code>\version</code>
<code>\category</code>	<code>\endrtrfonly</code>	<code>\mainpage</code>	<code>\remarks</code>	<code>\warning</code>
<code>\cite</code>	<code>\endverbatim</code>	<code>\manonly</code>	<code>\result</code>	<code>\weakgroup</code>
<code>\class</code>	<code>\endxmlonly</code>	<code>\memberof</code>	<code>\return</code>	<code>\xmlonly</code>
<code>\code</code>	<code>\enum</code>	<code>\msc</code>	<code>\returns</code>	<code>\xrefitem</code>
<code>\cond</code>	<code>\example</code>	<code>\mscfile</code>	<code>\retval</code>	<code>\\$</code>
<code>\copybrief</code>	<code>\exception</code>	<code>\n</code>	<code>\rtfonly</code>	<code>\@</code>
<code>\copydetails</code>	<code>\extends</code>	<code>\name</code>	<code>\sa</code>	<code>\ </code>
<code>\copydoc</code>	<code>\f\$</code>	<code>\namespace</code>	<code>\section</code>	<code>\&amp;</code>
<code>\copyright</code>	<code>\f </code>	<code>\nosubgrouping</code>	<code>\see</code>	<code>\~</code>
<code>\date</code>	<code>\f </code>	<code>\note</code>	<code>\short</code>	<code>\&lt;</code>
<code>\def</code>	<code>\f{</code>	<code>\overload</code>	<code>\showinitializer</code>	<code>\&gt;</code>
<code>\defgroup</code>	<code>\f}</code>	<code>\package</code>	<code>\since</code>	<code>\#</code>
<code>\deprecated</code>	<code>\file</code>	<code>\page</code>	<code>\skip</code>	<code>\%</code>
<code>\details</code>	<code>\fn</code>	<code>\par</code>	<code>\skipline</code>	<code>\"</code>
<code>\dir</code>	<code>\headerfile</code>	<code>\paragraph</code>	<code>\snippet</code>	<code>\.</code>
<code>\dontinclude</code>	<code>\hideinitializer</code>	<code>\param</code>	<code>\struct</code>	<code>\::</code>
<code>\dot</code>	<code>\htmlinclude</code>	<code>\post</code>	<code>\subpage</code>	
<code>\dotfile</code>	<code>\htmlonly</code>	<code>\pre</code>	<code>\subsection</code>	

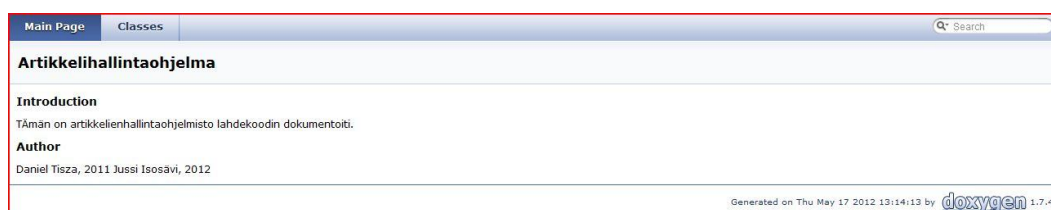
Työ on pienimuotoinen, joten listassa olevista avainsanoista on käytetty vain tarpeellisimpia. Aina pitää varoa liiallista koodin kommentointia, koska se voi haitata selkeyttä ja ylläpitoa. Alla olevassa taulukossa on listattu kaikki käytetyt avainsanat.

**Taulukko 4.** Työssä käytetyt avainsanat. /6/

Avainsana	Kohde	Käyttötarkoitus
\author	luokat	ilmoittaa tekijän
\brief	funktio, luokka	funktion ja luokkan kuvaus lyhyt kuvaaminen
\class	luokat, header, header name	tämä tunniste ilmoittaa doxygenille, että kommentti-lohkoon olisi liitetty luokka.
\details	lisätietoja	lisätietoja lohkon käyttöön liittyen
\mainpage	etusivun tiedot	tämän komenttilohkon sisältö laitetaan etusivulle
\param	funktiot eli metodit	metodin ja konstruktorin parametrin kuvaus
\section	luokka	luo jakson otsikon sivulle
\return	funktiot eli metodit	ilmoittaa palautusarvon

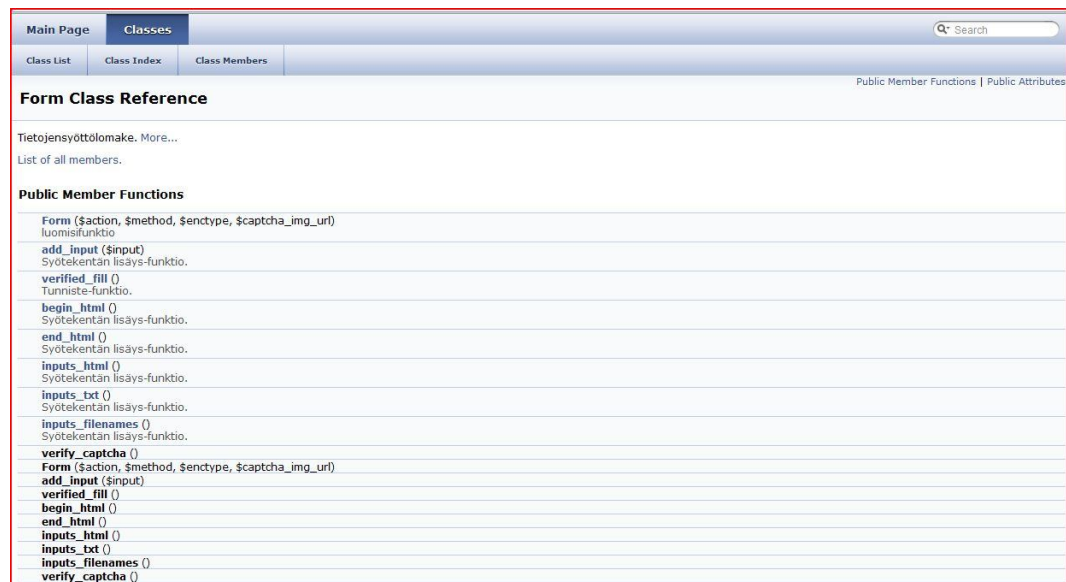
### 7.3 Dokumentoinnin tulos

Doxygen luo toisiinsa linkitetyt HTML-sivut, joissa on dokumentaatio.



**Kuva 41.** Dokumentaation aloitussivu.

Dokumentaatiota selataan seuraamalla linkkejä. Yksi sivu näyttää yhden kokonaisuuden.



**Kuva 42.** Esimerkinäkymä kommenteista.

Dokumentaatiosta voidaan myös hakea avainsanoilla tietoa esimerkiksi yksittäisestä funktiosta. Tämä helpottaa suuren dokumentaation hahmottamista.



**Kuva 43.** Dokumentaatiosta haku avainsanalla.

Seuraavaksi on esimerkki yhden funktion dokumentoinnista lähdekoodissa ja siitä muodostetusta dokumentaatiosta.

```
/*!  
 * \brief luomisifunktio  
 * \details luodaan uusi syötekenttä  
 *  
 * \param name käyttäjälle näytettävä nimi  
 * \param type käyttäjälle näytettävä tyyppi (numero/teksti)  
 * \param value käyttäjälle näytettävä oletusarvo  
 * \param size kentän koko  
 * \param errmsg käyttäjälle näytettävä virheteksti, jos arv ei kelpaa  
 *  
 */  
function Input( $name, $type, $value, $size, $errmsg )
```

**Kuva 44.** Funktion dokumentointi lähdekoodissa.

```
Input::Input ( $ name,  
              $ type,  
              $ value,  
              $ size,  
              $ errmsg  
              )
```

luomisifunktio  
luodaan uusi syötekenttä

**Parameters:**

<b>name</b>	käyttäjälle näytettävä nimi
<b>type</b>	käyttäjälle näytettävä tyyppi (numero/teksti)
<b>value</b>	käyttäjälle näytettävä oletusarvo
<b>size</b>	kentän koko
<b>errmsg</b>	käyttäjälle näytettävä virheteksti, jos arv ei kelpaa

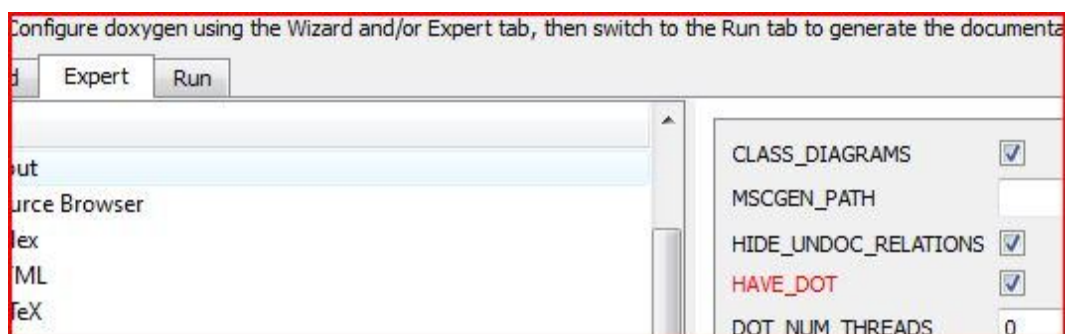
**Kuva 45.** Doxygenin generoima HTML-sivu funktiosta.



## Graphviz-ohjelman asennus

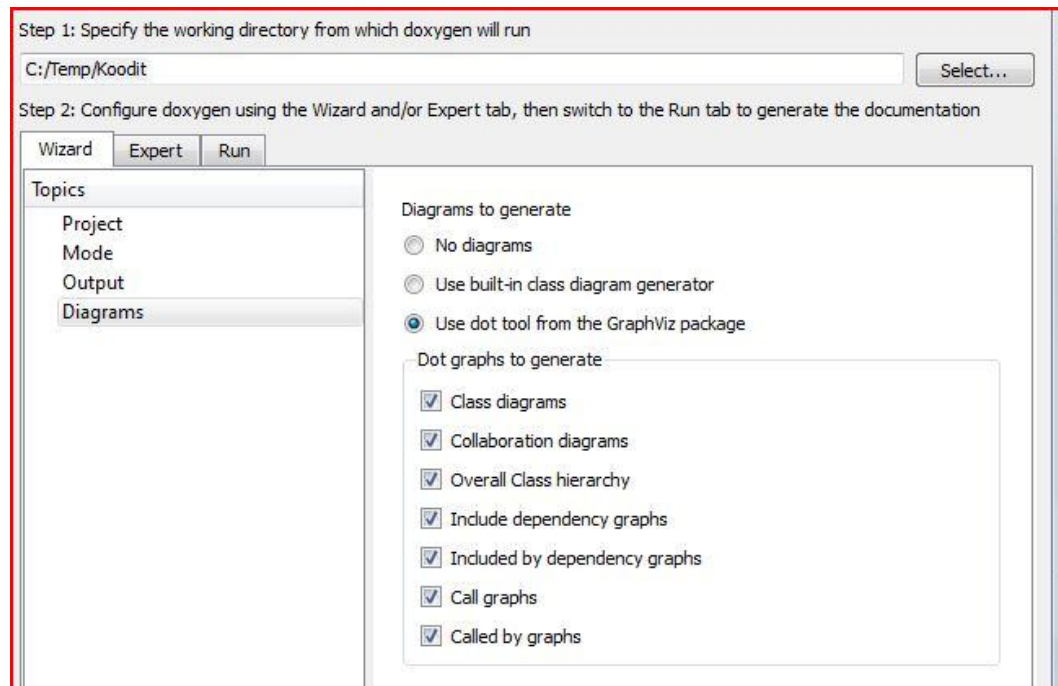
Graphviz 2.28 on Doxygenin käyttämä ohjelma, jolla voidaan luoda diagrammeja ja kaavoita. Se pitää asentaa kaikille käyttäjille. Sen voi ladata osoitteesta [http://www.graphviz.org/Download\\_windows.php](http://www.graphviz.org/Download_windows.php).

Graphviz asennettiin koneelle. Tämän jälkeen piti Doxygenin expert-asetuksista merkitä HAVE\_DOT -valinta päälle /1/.



**Kuva 46.** DOT-työkalun käyttöönotto Doxygenille.

Doxygen ohjelmassa pitää tehdä asetusmuutoksia, jotta diagrammien generointi onnistuu. Ohjelman koodi siirrettiin kansioon, jossa ei ole skandinaavisia merkkejä (C:/Temp/Koodit/). Tämän jälkeen kaikki määrittelyvalikon vaihtoehdot otettiin käyttöön.



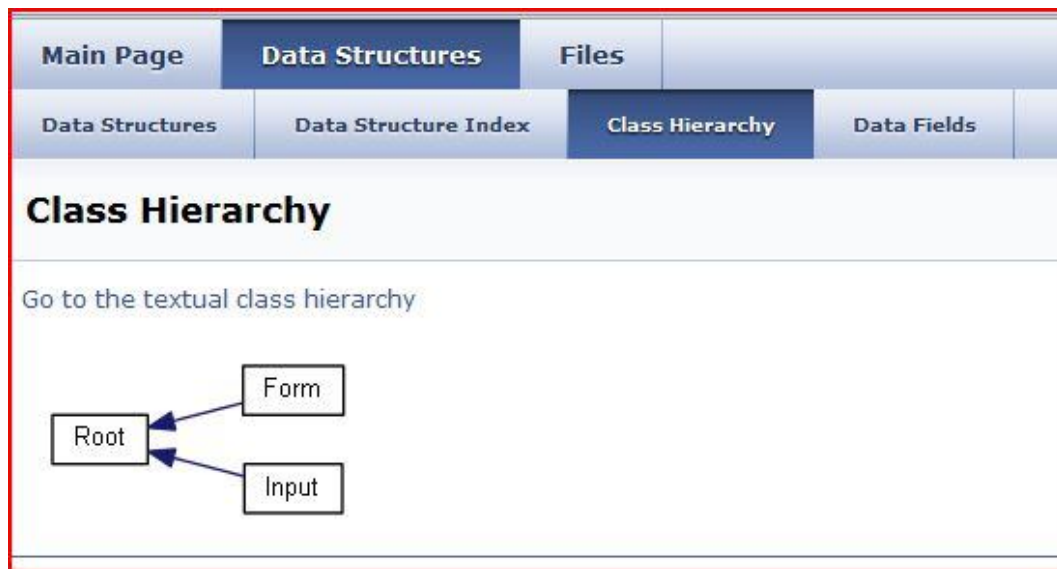
**Kuva 47.** Valitaan kaikki kuvaajat.

Alla olevassa kuvassa on kaikki tarvittavat määrittäykset kaavioiden luontiin.

```
#-----  
# Configuration options related to the dot tool  
#-----  
CLASS_DIAGRAMS      = YES  
MSCGEN_PATH         =  
HIDE_UNDOC_RELATIONS = YES  
HAVE_DOT           = YES  
DOT_NUM_THREADS    = 0  
DOT_FONTNAME       = Helvetica  
DOT_FONTSIZE       = 14  
DOT_FONTPATH       =  
CLASS_GRAPH        = YES  
COLLABORATION_GRAPH = YES  
GROUP_GRAPHS       = YES  
UML_LOOK           = YES  
TEMPLATE_RELATIONS = NO  
INCLUDE_GRAPH      = YES  
INCLUDED_BY_GRAPH  = YES  
CALL_GRAPH         = YES  
CALLER_GRAPH       = YES  
GRAPHICAL_HIERARCHY = YES  
DIRECTORY_GRAPH    = YES  
DOT_IMAGE_FORMAT   = png  
DOT_PATH           = "C:/Muut/Graphviz 2.28/bin"  
DOTFILE_DIRS       =  
MSCFILE_DIRS       =  
DOT_GRAPH_MAX_NODES = 50  
MAX_DOT_GRAPH_DEPTH = 20  
DOT_TRANSPARENT    = NO  
DOT_MULTI_TARGETS  = NO  
GENERATE_LEGEND    = YES  
DOT_CLEANUP        = NO
```

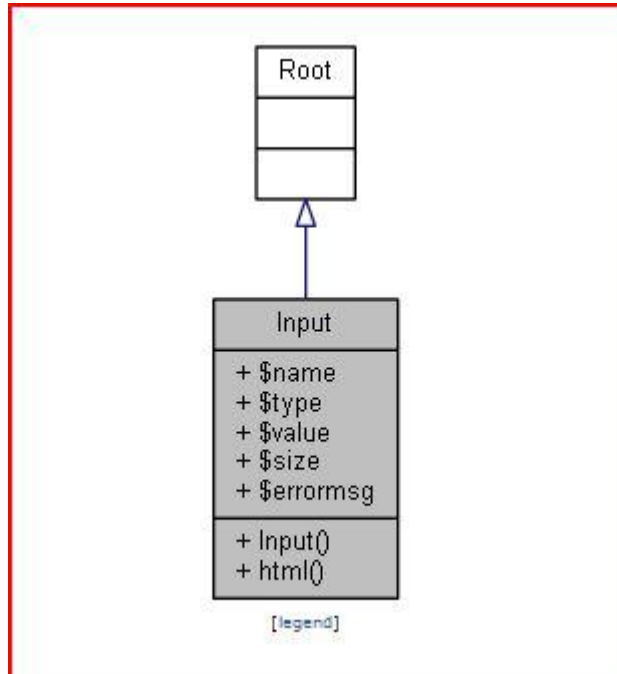
**Kuva 48.** Doxygenin asetukset kaavion tekoa varten.

Luokkakaavio on mahdollista generoida Doxygenilla käyttäen Graphviz-ohjelmaan kuuluvaa DOT-ohjelmaa.



**Kuva 49.** Generoitu luokkakaavio.

Kuvassa näkyy Root-luokasta periytyneet kaksi luokkaa From ja Input.



**Kuva 50.** Generoitu periytymiskaavio.

## 8 YHTEENVETO

Työssä paneuduttiin PHP-kielisen ohjelman dokumentointiin ja toteutukseen. Tässä ei ole syvennetty varsinaiseen ohjelmointiin. Se oli tarkoitus jättää vähiin. Kuitenkin ohjelmointia piti opiskella lisää, jotta ymmärsi paremmin koodin toimintaa ja rakennetta. Sen jälkeen saattoi tehdä lähdekoodiin tarpeelliset merkinnät, jotta järjestäjälle ja osallistujalle näkyvä informaatio saadaan sopivaan muotoon.

Dokumentointityöni alkuvaiheessa yrityksen saama toimeksianto peruuntui yliopiston järjestelyitten takia. Tämä lamaannutti työntekoani aikansa. Heräsi kysymys, voidaanko työtä jatkaa kyseisestä aiheesta. Yrityksen kanssa käydyissä keskusteluissa päädyttiin siihen, että lopputyönä ohjelmiston dokumentaatio saatetaan päätökseen alkuperäisen tavoitteen pohjalta. Toukokuun ensimmäisellä viikolla ilmeni, että yliopiston palvelimen asetuksia oli muutettu. Mahdollisesti kyseessä oli tietoturvaan liittyvä varmistus. Sovittiin, että käytetään yrityksen omistajalle kuuluvaa yliopiston sisäistä sähköpostiosoitetta testauksen ajan. Tähän liittyvissä kuvissa 24 ja 25 olevat osoitekentät on jätetty tyhjiksi tietoturvan takia.

Wamp server 2 -ohjelmiston vahvan tietoturvan takia syntyi ongelmia. PHPMyAdmin-sivulle pääsyyn saatiin muokatuksi oikeudet. Jostakin syystä luotua ilmoittautuneet-tietokantaa ei ollut tietokantalistassa, joten sitä ei voinut käyttää normaaliin tapaan, vaikka oikeudet tietokannan muokkaukseen lopulta aukenivat. Tässä ongelma ratkaistiin ottamalla käyttöön XAMPP-palvelin, joka toimi hyvin. Wamp Server 2 -ohjelmaa en suosittelen käytettäväksi asennuksen jälkeen ilmenneiden monien hankaluuksien takia.

Loin PHP-koodin, joka listaa osallistajat ilmoittautuneet-tietokannasta. Se sijaitsee palvelimen sovelluskerroksessa ja koodi listaa ne www-sivulle.

Yrityksen kannalta työn tulos jää odottamaan hyödyntämistä. Jatkokehittelyyn kannattaa ryhtyä, jos yritys saa uusia tilauksia, jotka soveltuvat artikkelienhallintasovelluksen rakenteisiin. Sellaisia voisivat olla messutapahtumien järjestelyt tai

suuren yksittäisen yrityksen tuotteiden markkinoinnissa potentiaalisten ostajien kutsuminen ja jopa alihankkijoiden kilpailutustilaisuuksiin houkuttelu.

Myöhemmissä versioissa ohjelmaa voisi kehittää monella tavalla. Käyttöliittymä voidaan kääntää usealle kielelle ja sivun ylälaitaan sijoittaa valintavaihtoehtoiksi esimerkiksi maiden liput. Osallistujan on mahdollista rekisteröityä palveluun omalla salasalla ja sen jälkeen suorittaa ilmoittautuminen. Tämä auttaa käyttäjää, jos on tarkoitus käydä useissa tapahtumissa. Listausta voisi vielä kehitellä joustavammaksi siten, että käyttäjä voi aloitussivulla tarvittaessa myös itse poistaa osallistumisensa konferenssiin. Ilmoittautumisessa tunnistekuva ei tällä hetkellä sovellu näkörajoitteisille ihmisille. Tällöin voidaan käyttää hyväksi ääntä.

Työssä oli varsin hankala päästä alkuun. Tietokantoihin ja PHP-ohjelmointiin perehtyminen vei aikaa viikkokausia. PHP-kielen aikaisempi tuntemus helpotti dokumentointia ja samalla auttoi muidenkin tässä käytettyjen ohjelmointikielten ymmärtämisessä. Materiaalia kertyi runsaasti ja sitä piti karsia jonkin verran. Dokumentointityökalun käyttöohjeen siirsin opinnäytetyön liitteeksi. Ohje antaa yleiskuvan työkalun käyttöliittymästä sekä näyttää lyhyesti dokumentaation generoinnin vaiheet. Ohjelmoinnin saloista pääsin ehkä alkuportaita ylemmäs.

## LÄHTEET

- /1/ Doxygen Manual Viitattu 9.5.2012.  
<<http://www.star.bnl.gov/public/comp/sofi/doxygen/>>
- /2/ Gilmore, W. J. PHP 5 & MySQL.- Tehokas hallinta (2004). Gummerus Kirjapaino Oy, Jyväskylä.
- /3/ Haikala, I. Märijärvi, J. (2004). Ohjelmistotuotanto. Karisto Oy, Hämeenlinna.
- /4/ van Heesch, D. (1997-2011). Doxygen. Viitattu 15.1.2012.  
<<http://www.stack.nl/~dimitri/doxygen/features.html>>
- /5/ van Heesch, D. (1997-2011). Doxygen. Viitattu 15.1.2012.  
<<http://www.stack.nl/~dimitri/doxygen/index.html>>
- /6/ van Heesch, D. (1997 - 2011). Doxygen. Viitattu 27.2.2012.  
<<http://www.stack.nl/~dimitri/doxygen/commands.html#cmdauthor>>
- /7/ van Heesch, D. (1997 - 2012). Viitattu 20.3.2012. Päivitetty 28.2.2012.  
<http://www.stack.nl/~dimitri/doxygen/projects.html>
- /8/ van Heesch, D. (1997-2011). Doxygen. Viitattu 23.3.2012.  
<<http://www.stack.nl/~dimitri/doxygen/starting.html>>
- /9/ van Heesch, D. (1997-2011). Doxygen. Viitattu 10.4.2012  
<<http://www.stack.nl/~dimitri/doxygen/docblocks.html>>
- /10/ Internetix. Pihl, K. (1998). Viitattu 20.3.2012.  
<<http://oppimateriaalit.internetix.fi/fi/avoimet/atk/tietokanta/tkopi3/sql>>
- /11/ Jyväskylän ammattikorkeakoulu. Viitattu 28.4.2012.  
<[http://batman.jamk.fi/~tuito/www\\_sk/phpperusteet.htm](http://batman.jamk.fi/~tuito/www_sk/phpperusteet.htm)>

- /12/ Koulutuskeskus Salpaus. Viitattu 6.5.2012.  
<[http://edu.phkk.fi/Opiskelu/sqlkieli/sql\\_osa1.htm](http://edu.phkk.fi/Opiskelu/sqlkieli/sql_osa1.htm)>
- /13/ MySQL Official Page. Viitattu 14.5.2012.  
<http://dev.mysql.com/doc/refman/5.0/fr/insert.html?ff=nopfpls>
- /14/ MySQL Official Page. Viitattu 30.4.2012.  
<<http://mysql.com/why-mysql/>>
- /15/ Palo Alto Research Center. Viitattu 20.3.2012.  
<<http://www2.parc.com/istl/projects/captcha/history.htm>>
- /16/ PHP Official Page. Viitattu 28.4.2012.  
<[http://fi.php.net/manual/en/faq\\_general.php](http://fi.php.net/manual/en/faq_general.php)>
- /17/ PHP.net - History of PHP. Viitattu 21.5.2012.  
< <http://php.net/manual/en/history.php.php>>
- /18/ TechNet Magazine - Use Msconfig to Disable and Enable UAC.  
Viitattu 6.5.2012,  
<<http://technet.microsoft.com/en-us/magazine/dd296719.aspx>>
- /19/ The New York Times. Viitattu 10.2.2012. Saatavilla Internetissä:  
<<http://www.nytimes.com/2002/12/10/science/human-or-computer-take-this-test.html>>
- /20/ The Official CAPTCHA Site. Viitattu 18.2.2012.  
<<http://www.captcha.net/>>
- /21/ Viescas, J. (2000). Microsoft Access 2000 - Käyttöohjekirja. Jyväskylä. IT Press. Gummerrus Kirjapaino Oy.
- /22/ XAMPP Official Page. Viitattu 6.5.2012.  
<<http://www.apachefriends.org/en/xampp.html>>