



LAUREA
AMMATTIKORKEAKOULU

Uuden edellä

Ohjelmistokehityksen käyttö web-ohjelmoinnissa - Case listaa.fi

Taimén, Hanna

2012 Leppävaara

Ohjelmistokehyksen käyttö web-ohjelmoinnissa - Case listaa.fi

Taimén, Hanna

Ohjelmistokehityksen käyttö web-ohjelmoinnissa - Case listaa.fi

Vuosi 2012 Sivumäärä 29

Tämän toiminnallisen opinnäytetyön tavoitteena oli toteuttaa Agilo Partners -yritykselle tietokantapohjainen verkkosivusto, jossa käyttäjät voivat vaivattomasti ja nopeasti jättää ilmoituksia myytävistä, vaihdettavista, ostettavista tai vuokrattavista tavaroista tai selata muiden vastaavanlaisia ilmoituksia.

Sivuston ulkoasuunitelma, sivuston ilme sekä toimintaperiaate tulivat toimeksiantajayritykseltä, mutta ohjelmointikieltä sekä -alustaa ei rajattu. Ilmoitussivuston ohjelmointikieleksi valikoitui PHP-ohjelmointikieli sekä MySQL-tietokanta, sillä se on hyvin yleinen yhdistelmä Internet-kehityksessä, ja lisäksi PHP-kieli oli entuudestaan tuttu.

Opinnäytetyön pääasiallisena tarkoituksena oli luoda toiminnallinen, nykyaikainen sekä tietoturvallinen ilmoitussivusto tehokkaasti. Tämän vuoksi sivusto toteutettiin käyttäen Yii-ohjelmistokehystä, jonka avulla sivusto saatiin luotua nopeasti. Yii-ohjelmistokehys soveltui tämänkaltaisen pienen, mutta vuorovaikutteisen sivuston toteutukseen erinomaisesti, sillä ohjelmistokehys huomioi yleisimmät tietoturvallisuusriskit. Lisäksi Yii hyödyntää Model-View-Controller -mallia, jonka avulla ohjelmointi on tehokasta ja sujuvaa.

Opinnäytetyön lopputuloksena syntyi selkeä verkkosivusto, joka vastaa toiminnaltaan toimeksiantajan vaatimuksia. Työssä myös todetaan Yii-ohjelmistokehityksen toimivan hyvin pienen ohjelmistokehitysprojektin alustana.

Asiasanat internet-sivusto, web-sovellukset, ohjelmistokehitys

Sisällys

1	Johdanto	7
1.1	Toimeksiantaja.....	7
1.2	Aiheen rajaus.....	7
1.2.1	Toiminnallisen työn rajaus.....	8
1.2.2	Oppimistavoite.....	8
1.3	Menetelmät.....	8
1.4	Ohjelmointikielen ja työkalujen valinta.....	8
1.5	Opinnäytetyön menetelmä.....	9
1.6	Opinnäytetyön tavoitteet.....	9
1.7	Sivuston kohderyhmä.....	9
2	Internet-sivujen toiminta.....	9
2.1	HTML-kieli.....	10
2.2	CSS-kieli.....	11
2.3	PHP:n toimintaperiaate.....	11
2.4	MySQL:n toimintaperiaate.....	11
2.5	Toiminnallisuuden lisääminen sivustolle.....	11
2.5.1	JavaScript.....	12
2.5.2	jQuery.....	12
2.5.3	AJAX.....	12
3	Yleiset tietoturvariskit Internet-sivuilla.....	13
4	Pienen ohjelmistokehitysprojektin malli.....	13
4.1	Määrittely.....	14
4.2	Suunnittelu.....	14
4.3	Toteutus.....	14
4.4	Testaus ja käyttöönotto.....	14
5	Listaa.fi -sivuston työn kuvaus.....	14
5.1	Suunnittelu.....	15
5.1.1	Tietokantasuunnittelun teoria.....	15
5.1.2	Listaa.fi sivuston tietokantakaavio.....	15
5.1.3	Tietokannan rakentaminen.....	16
5.2	Käyttöliittymä ja käytettävyys.....	17
5.3	Käyttöliittymä listaa.fi:ssä.....	17
5.4	Toteutus ohjelmistokehyksen avulla.....	18
5.4.1	Teoria Yii:stä.....	18
5.4.2	MVC-malli.....	19
5.4.3	Hyödyt ohjelmistokehyksestä.....	20
6	Työn lopputulos.....	21
6.1	Etusivu.....	21
6.2	Ilmoitusten selaus.....	22
6.3	Ilmoitussivu.....	22
6.4	Yhteenveto.....	24
7	Loppusanat.....	25

Lähteet..... .27
Kuvat..... .29

1 Johdanto

Opinnäytetyön toimeksiantona on suunnitella ja toteuttaa tietokantapohjainen verkkopalvelu ”listaa.fi” start up -yritykselle, Agilopartners Oy:lle. Sivustolla käyttäjät voivat jättää ilmoituksia vaivatta, ilman rekisteröitymistä sekä ilman kuluja. Ilmoitukset voivat olla myynti-, vaihto-, vuokra- tai ostoilmoituksia. Sivustolla voi nopeasti etsiä ilmoituksia eri kategorioittain valitsemalta alueelta Suomessa, esimerkiksi Uudeltamaalta. ”Listaa.fi”-sivuston tarkoituksena on kilpailla suosituimpien ilmoitussivustojen kanssa, kuten ”tori.fi”-sivuston.

1.1 Toimeksiantaja

Toimeksiantajayritys on pieni, aloitteleva yritys Agilopartners Oy. Yritys perustaa toimintaansa erilaisille ilmoitussivustoille - tavoitteena on mahdollistaa käyttäjille mahdollisimman helppo tapa ilmoittaa kauppatavaratuotteita Internetissä. Yritys on jo perustanut ”vuokraforum.fi”-sivuston, jossa käyttäjät voivat etsiä vuokrattavia asuntoja tai ilmoittaa omista vuokrattavista asunnoistaan. ”Nettiformum”-perheeseen on lisäksi kehitteillä ”asuntoforum”- ja ”autoforum”-sivustot. Yritys on nähnyt ilmaisten ilmoitussivustojen suosion ja tarpeen Suomessa, ja ovat lähteneet kehittämään omia kilpailevia sivustojaan markkinoille. ”Listaa.fi” ei varsinaisesti kuulu ”Nettiformum”-perheeseen, mutta idealtaan ja toimintaperiaatteeltaan se on samansuuntainen.

1.2 Aiheen rajaus

Työssä tarkastellaan pienen ohjelmistotyön rakennetta ja suunnittelua ennen varsinaisen toteutuksen alkamista. Asiakkaan, eli toimeksiantajayrityksen, toiveet tulee huomioida määrittelyvaiheessa ja ehdottaa niihin parannuksia teorioiden sekä omien näkemysten perusteella. Kirjallisen työn pääpaino tulee olemaan ohjelmistokehyksen (engl. framework) ja avointen ohjelmistokirjastoiden hyödyntämisessä web-ohjelmoinnissa. Työssä tutkitaan ohjelmistokehyksien tuomia etuja verraten siihen, että ohjelmointityö tehtäisiin tyhjän päälle. Kirjallisen työn rajaus kohdentuu lyhyeen teoriaviitekehykseen, toimintaprosessin eli sivuston rakentamisen kuvaamiseen, sekä ohjelmistokehyksien ja -kirjastojen hyödyntämisen syventymiseen. Pääpaino kirjallisella työllä on olla toimintaprosessin tukena ja kirjallisena raportointina. Opinnäytetyössä pyritään vastaamaan kysymyksiin, mitä etuja ohjelmistokehyksen käyttämisestä on pienessä web-ohjelmointiprojektissa sekä miten pieni projekti toteutetaan tehokkaasti.

1.2.1 Toiminnallisen työn rajaus

Sivuston teko on rajattu opinnäytetyön osalta niin, että käyttöliittymäsuunnittelu jää pienempään osaan, sillä toimeksiantaja tekee piirrokset elementeistä ja sivuston ulkoasusta. Kirjallisuustutkimuksen pohjalta tarkoituksena on yhteistyössä toimeksiantajayrityksen kanssa tehdä mahdollisimman nykyaikainen ja toimiva käyttöliittymä, joten parannuksia ja ehdotuksia pyritään löytämään.

1.2.2 Oppimistavoite

Opinnäytetyön aihe on valittu huolella, sillä web-palveluiden tuottaminen on tärkeä taito nykypäivänä. Opinnäytetyössäni aion kehittää tehokasta ohjelmointitaitoa sekä oppia lisäämään interaktiivisuutta Internet-sivustolle. Sivuston on oltava nykyaikainen niin käyttöliittymältä kuin toiminnallisuudeltaankin, jotta sivusto kiinnostaa käyttäjiä. Henkilökohtaisena tavoitteena on oppia ohjelmoimaan dynaamisia internet-sivustoja ammattimaisesti ja tehokkaasti.

1.3 Menetelmät

Tämä työ on toiminnallinen opinnäytetyö, jossa pääpaino on tekemällä oppimisessa. Opinnäytetyössä ei tehdä tarkkaa kirjallisuustutkimusta, vaan kirjallisuuskatsausosuudella pyritään tukemaan ”listaa.fi”-sivuston työstämistä ja siinä tehtyjä valintoja. Kirjallisessa työssä kuvataan prosessi sekä siihen käytettyjä teorioita ja analysoidaan käytettyjen työmenetelmien toimivuutta pienehkössä ohjelmistokehityksessä.

1.4 Ohjelmointikielen ja työkalujen valinta

Sivusto toteutetaan HTML-kielellä, ja sivuston dynaamiseen ohjelmointiin käytetään PHP-ohjelmointikieltä. Toimeksiantaja ei ottanut kantaa ohjelmointikielen valintaan, vaan he antoivat siihen vapaat kädet. PHP vaikoitui ohjelmointikieleksi siksi, että siitä oli ennestään kokemusta ja tämän opinnäytetyön tekijä päätti vahvistaa taitoja sen sijaan, että olisi valinnut uuden kielen opeteltavaksi. Työn ohjelmoinnissa käytetään lisäksi Yii-ohjelmistokehystä, jonka valikoitui sen korkean suorituskyvyn vuoksi (Yii Software LLC, 2011). ”Listaa.fi” -sivuston tietokantana käytetään MySQL-tietokantahallintajärjestelmää, joka on toimeksiantajayrityksen valinta.

Sivuston ohjelmointiin käytetään Netbeans-ohjelmointiympäristöä, koska siitä on entuudessaan kokemusta, ja se on todettu hyväksi ohjelmaksi projektin toteutukseen.

Projektinhallintatyökaluna projektissa toimii GoPlan-internetsovellus (<http://goplanapp.com/>), jonne kaikilla projektin jäsenillä on oikeudet.

1.5 Opinnäytetyön menetelmä

Toiminnallisessa opinnäytetyössä ei välttämättä tarvitse käyttää tutkimusmenetelmiä, vaikka viitekehysten kerääminen tuleekin hakea harkitusti. Toiminnallisessa opinnäytetyössä on riski, että työstä tulee liian laaja, mikäli työssä on tutkimuksellinen kirjallisuusselvitys aiheesta. (Vilka & Airaksinen 2003, 56).

1.6 Opinnäytetyön tavoitteet

Työssä tuotettava verkkopalvelu, ”listaa.fi” on toimintakeskeisen opinnäytetyön päätavoite. Sen tarkoituksena on olla nykyaikainen verkkosovellus - niin ulkonäöllisesti kuin toiminnallisestikin. Sivusto pyritään suunnittelemaan ja toteuttamaan yhdessä toimeksiantajan kanssa niin, että se on viimeistä lopputestausta vaille valmis laitettavaksi esille Internetiin.

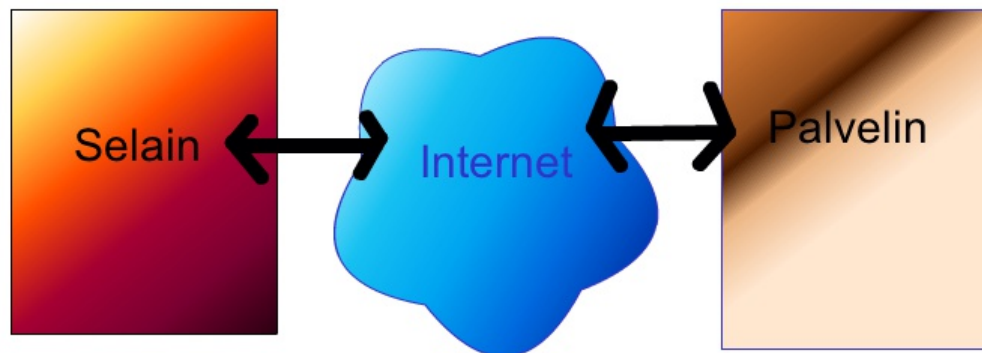
Nykyaikaisuus toteutettavalle verkkosivustolle rakennetaan uusien menetelmien, kuten jQuery-JavaScript-kirjaston avulla. Yii-ohjelmistokehystä käyttäen sivustosta saadaan tietoturvallinen. Sivuston ulkoasu tulee olemaan ilmava ja sivulla käytetään AJAX-ohjelmointia, jolloin sivustolla voidaan vaihtaa sisältöä lataamatta koko sivua uudelleen.

1.7 Sivuston kohderyhmä

”Listaa.fi”-verkkosivuston kohderyhmä on kaikki Internetin käyttäjät, jotka etsivät helppoa ja nykyaikaista ilmoituksen jättö -sivustoa. Heillä on tarve ilmoittaa jostain myytävästä tai ostettavasta tuotteesta tai he etsivät jotain ostettavaa. Sivuston tulee olla toiminnaltaan niin helppo ja yksinkertainen, että sen vuoksi kohderyhmä ei pienene. Sivuston päätavoite on olla nopeakäyttöinen ja toimiva.

2 Internet-sivujen toiminta

Perinteisten internet-sivujen toimintamalli on yksinkertainen asiakas-palvelin-malli. Asiakasohjelma eli web-selain lähettää pyyntöjä palvelimelle, jossa palvelinohjelma vastaa selaimen pyyntöihin. Yhteys asiakkaan ja palvelimen välillä tapahtuu HTTP-siirtoprotokollan yhteyskäytännön mukaan. Web-dokumentit toteutetaan HTML-kuvauskielellä.



Kuva 1: Internetin toiminta. Selain tekee pyynnön, jonka palvelin palauttaa käyttäjän selaimelle.

On kahdenlaisia Internet-sivustoja: staattisia ja dynaamisia. Staattinen sivusto on muuttumaton sivusto, jonka sisältöä voi muuttaa vain puuttamalla lähdekoodiin. Dynaamisia eli muuttuvia sivustoja voidaan kutsua web-sovelluksiksi. Ne perustuvat käyttäjien vuorovaikutukseen. Esimerkiksi Internet-huutokaupat ovat dynaamisia eli interaktiivisia sivustoja, jossa käyttäjät voivat muuttaa sivuston sisältöä muun muassa lomakkeiden avulla. Dynaamisen sivuston toimintamallissa palvelimen saatua pyyntö selaimelta, se lähettää kyselyn tietokantaan. Kun palvelin on saanut vastauksen tietokannasta, se palauttaa selaimelle vastauksen HTML-dokumenttina. (Rantala 2002, 4-7.)

2.1 HTML-kieli

HTML-kieli (HyperText Markup Language) on W3C:n ylläpitämä julkaisukieli, joka koostuu sivun rakennetta kuvaavista elementeistä sekä niiden attribuuteista. Internet-selaimet tulkitsevat HTML-dokumentteja hieman eri tavoin, joten siksi kaikki eivät näytä samalta jokaisessa selaimessa. (Keränen ym. 2006, 30.)

```

<!DOCTYPE html>
<html>
<body>

<h1>My First Heading</h1>
<p>My first paragraph.</p>

</body>
</html>
  
```

Kuva 2: HTML-koodin rakenne

2.2 CSS-kieli

CSS-kielillä (Cascading Style Sheet) laaditaan HTML-dokumenttiin ulkoasu luomalla tyyliohjeita. Dokumentin ulkoasu on helpommin muokattavissa, mikäli ulkoasu määritellään omassa css-dokumentissaan. CSS-kielillä voidaan esittää lähes mitä vain tyyliasetuksia, kuten elementin kokoja, värejä, kirjasimia tai sijainteja. Tyylitiedostoon kerätään tyyli luokkia. (Keränen ym. 2006, 33.)

2.3 PHP:n toimintaperiaate

PHP (Hypertext Preprocessor) on ohjelmointikieli, jonka komentoja voidaan kirjoittaa HTML-koodin väliin (Heinisuo & Rauta 2007). Rantalan (2002, 12) mukaan PHP:n tarkoituksena on olla apuvälineenä dynaamisten web-sivujen ja -sovellusten tekoon. PHP on tulkittava kieli, eli koodi ajetaan joka kerta juuri ennen, kuin web-palvelin lähettää sivuston selaimelle. Tulkittavia kieliä kutsutaan yleisesti Skriptikieliksi. PHP:n avulla käytetään tietokantoja, kuten MySQL-tietokantaa. PHP on avoimen lähdekoodin ohjelmointikieli.

2.4 MySQL:n toimintaperiaate

MySQL on tietokannanhallintajärjestelmä, jolla sanan mukaisesti hallitaan tietokantoja. Tietokannat ovat tietovarastoja, joihin esimerkiksi dynaamisten verkkosivujen tietoja tallennetaan. MySQL:n avulla tietoa haetaan tietokannasta tai tietoa tallennetaan tietokantaan. Tietokannoista yleisin on relaatiotietokanta, jossa tiedot tallennetaan tauluihin. Taulujen väliset suhteet ovat relaatioita. Niiden avulla taulujen tietoja voidaan yhdistää toisiin tauluihin. MySQL-tietokannanhallintajärjestelmää voidaan käyttää usean eri ohjelmointikielen, kuten PHP-, Python- ja C++ -kielten, kanssa. (Heinisuo & Rauta, 37-38.)

MySQL toimii asiakas-palvelin arkkitehtuurin mukaisesti, eli MySQL pyörii omana sovelluksenaan tietokoneella ja toimii palvelimena toisille ohjelmille. MySQL:n kyselykieli on SQL, mikä tarkoittaa sitä, että yhteys web-palvelimesta MySQL-tietokantapalvelimeen otetaan PHP:n lähettäessä SQL-kielisiä kyselyitä, joihin MySQL vastaa lähettämällä tietoa tietokannasta. Yhdellä tietokantapalvelimella voi olla useita tietokantoja, joiden sisällä on useita tauluja. (Heinisuo & Rauta, 39-40).

2.5 Toiminnallisuuden lisääminen sivustolle

Nykyään Internet-sivustot ovat monitoimintaisia sovelluksia. Näitä sovelluksia kutsutaan rikkaimiksi Internet-sovelluksiksi (engl. rich internet applications), ja niille on tyypillistä

interaktiivisuus käyttäjän ja sivuston välillä. Rikkaissa sovelluksissa vain osa sivusta ladataan uudelleen sivun päivittyessä sen sijaan, että koko sivu ladattaisiin uudestaan. Tyypillistä on myös reaaliaikainen palaute käyttäjän käyttäytymisestä sivustolla, kuten virheilmoitusten näyttäminen tilanteessa, jossa käyttäjä täyttää lomaketta jonkin tiedon ollessa puutteellinen. Ponnahdusikkunat ja sivun osittaminen ovat tyypillisiä elementtejä rikkaissa Internet-sovelluksissa. (Maurer 2006).

2.5.1 JavaScript

JavaScriptiä käytetään luomaan Internet-sivustoille asioita, joita HTML-kielellä ei saa aikaiseksi. Sillä saadaan interaktiivisuutta sivustolle; JavaScriptillä voidaan luoda interaktiivisia lomakkeita, suorittaa käyttäjän syöttämän tiedon validointia, reagoida käyttäjän hiiren tekemiin liikkeisiin välittömästi ja luomaan Internet-sivuista nykyaikaisemman näköisiä (Keogh 2005, Introduction). JavaScriptillä saadaan siis laajennettua Internet-sivujen toimintaa tai jotain sivun osaa. (Peltomäki 2001, 9).

JavaScript on tulkettava skriptikieli, joka upotetaan HTML-dokumentin väliin. Upotettava koodi voi olla pituudeltaan yhdestä rivistä eteenpäin aina kokonaisuun sovelluksiin asti. JavaScript-koodi tulkitaan rivi riviltä, ja se suoritetaan sen mukaisesti. JavaScriptin lukemiseen tarvitaan tulkkia, joka voidaan asentaa isäntäsovellukseen, kuten Internet-selaimeen tai -palvelimeen. (Peltomäki 2001, 9.)

2.5.2 jQuery

jQuery on JavaScriptin päälle kirjoitettu kirjasto, jonka avulla JavaScriptin kirjoittaminen on täysin erilaista, nopeaa ja ytimekästä. jQuery on suunniteltu muuttamaan JavaScript-koodin kirjoittamisen tyyliä helpompaan suuntaan. jQuery:n on luonut voittoa tavoittelematon säätiö vuonna 2009. Se on edelleen omistautunut kehittämään jQuery:n kirjastoja, joihin lukeutuu peruskirjaston lisäksi käyttöliittymää kehittävä jQuery UI - sekä mobiiliselaimille sopiva jQuery Mobile -kirjastot. (jQuery Foundation).

2.5.3 AJAX

AJAX (Asynchronous Javascript and XML) on yleisnimitys monen tekniikan yhteiskäytännölle. Sillä tehdään asioita internetsovelluksissa, joita pelkällä Javascript-koodilla ei voida tehdä. AJAX:n avulla internetsovellukset saadaan näyttämään perinteisiltä tietokoneohjelmilta, kun sovelluksen näkymät vaihtuvat ilman sivun uudelleenlataamista. Näin ollen internetsovellus on entistä dynamisempi. AJAX:ia käytetään kirjoittamalla Javascript-sovellus, joka kutsuu

samaa www-palvelinta XMLHttpRequestin avulla. Helpoin tapa käyttää AJAX:ia on käyttää jQuery-kirjastoa. (Ekonoja ym. 2010).

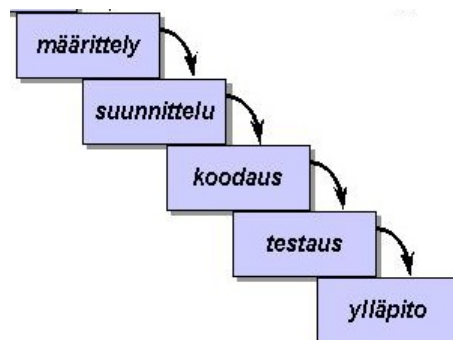
3 Yleiset tietoturvariskit Internet-sivuilla

Hakkeri on henkilö, joka käyttää tietotekniikan tuntemustaan johonkin laittomaan tai haitalliseen toimintaan. Useimmiten hakkeri murtautuu johonkin tietojärjestelmään, jakaa viruksia tai yleisesti häiritsee tietojärjestelmien toimintaa. Hakkerin yleisempiä syitä teoilleen on utelaisuus tietokoneiden toimintaa kohtaan, ilkeä tai jokin haaste. Hakkerit voivat toimia yksin tai porukassa. (Ruohonen, M.2002, 320-321.)

Verkkosivujen ohjelmistokoodissa voi olla tietoturva-aukkoja, joiden avulla hakkeri suorittaa komentoja. Yleisin tilanne on interaktiivisilla sivuilla, joilla on esimerkiksi lomakkeita. Kun käyttäjä syöttää lomakkeelle tiedon, joka menee tietokannan käsiteltäväksi, täytyy käyttäjän syöttämä data tarkistaa ja varmentaa. Jos sitä ei tarkisteta, voi siinä olla ohjelmaa suorittavia komentoja mukana, joilla hakkeri murtaa verkkosivut. (Ruohonen, M. 2002. 358.)

4 Pienen ohjelmistokehitysprojektin malli

”Listaa.fi”-sivuston luonnissa on kyse projektista, jossa toimeksiantajalla on selkeä visio, minkälainen sivustosta tulee. Näin ollen varsinaista ketterää ohjelmistokehitysprosessia ei tarvita, vaan projekti etenee suoraviivaisesti kohti määränpäättä. Tässä projektissa on käytetty kaikista vanhinta ohjelmistoprojektin mallia, eli vesiputousmallia. Se on lineaarinen malli, jossa jokaisella vaiheella on selkeä alku ja päämäärä, jonka tavoittamisen jälkeen alkaa uusi vaihe. (Mohapatra, P. 2010, 19).



Kuva 3: Esimerkki vesiputousmallista

4.1 Määrittely

Vesiputousmallin ensimmäinen vaihe on vaatimusmäärittely. Siinä käydään asiakkaan kanssa läpi projektin tavoite, ja millaiseen lopputulokseen projektissa pyritään. Tämän vaiheen jälkeen tekijöillä on oltava tiedossa, mitkä ovat projektin tavoitteet ja mitä osa-alueita siinä on. (Mohapatra, P. 2010, 18-25.)

4.2 Suunnittelu

Suunnitteluvaihe sisältää projektin yksityiskohtaisen suunnittelun asiakkaan kanssa tehdyn vaatimusmäärittelyn perusteelta. Suunnitteluvaiheessa suunnitellaan sovelluksen tietokantarakenne, ohjelmistokoodin rakenne, työn ulkoasu sekä projektin vaiheet. Hyvin suunniteltu työ helpottaa toteutusvaihetta. Mohapatra, P. 2010, 18-25.)

4.3 Toteutus

Toteutusvaihe on työn toiminnallinen osuus. Siinä luodaan tietokanta, asennetaan ohjelmistokehys, kirjoitetaan koodia ja rakennetaan projekti toiminnalliseksi. Jos vesiputousmallin aiemmat vaiheet on tehty huolellisesti, vie toteutus vain pienen osan koko projektiin kulutetusta ajasta. (Mohapatra, P. 2010, 18-25).

4.4 Testaus ja käyttöönotto

Sovelluksen toteutustyön jälkeen on testauksen aika. Testauksen tarkoituksena on löytää sovelluksesta virheet, jotta ne voidaan korjata ennen julkaisua. Kun sovellus tavoittaa tilan, jossa sillä ei ole enää suuria virheitä, voidaan se julkaista ja ottaa käyttöön. (Haikala & Märijärvi, 153.)

Tässä opinnäytetyössä lopputestaus ja käyttöönotto jätetään jatkokehityksen työstettäväksi. Se on suuri kokonaisuus, johon kuuluu sivuston ylläpito.

5 Listaa.fi -sivuston työn kuvaus

Tässä kappaleessa kuvataan listaa.fi-sivuston työn kuvaus nojaten siihen pohjautuvaan teoriaan. Sivusto on toteutettu vesiputousmallilla toki laatua varmistaen koko prosessin ajan.

5.1 Suunnittelu

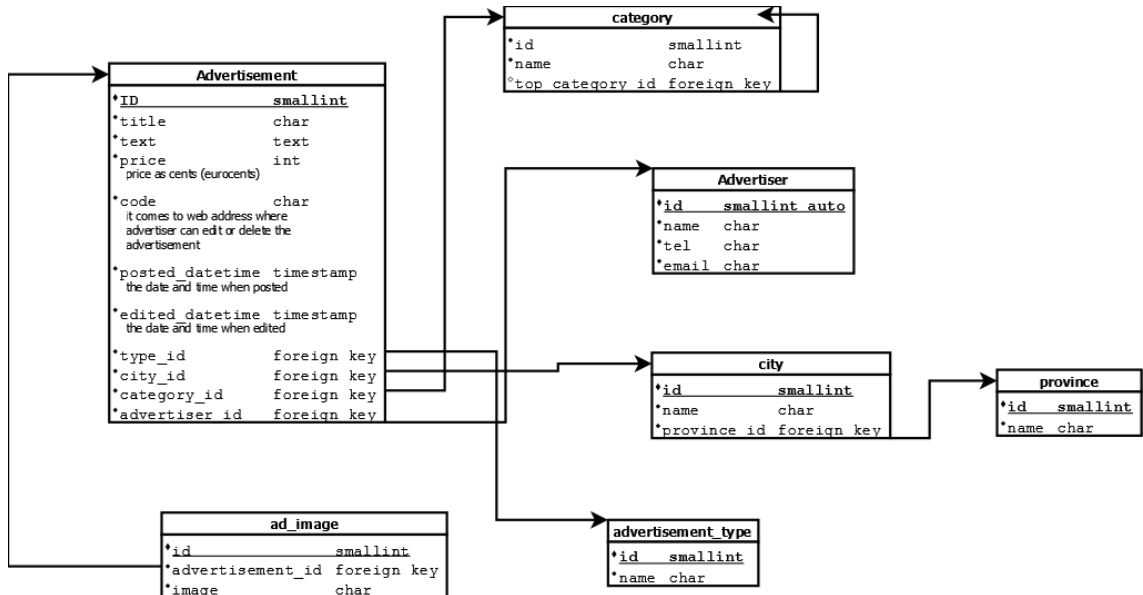
5.1.1 Tietokantasuunnittelun teoria

Relaatiotietokanta sisältää muuttuvaa tai käytössä olevaa tietoa, joka esitetään useissa tauluissa, joissa on rivejä ja sarakkeita. Tarvittava tieto pyritään tallentamaan relaatiotietokantaan niin, että yksi tieto löytyy vain yhdestä paikasta. Tietokantaan tallennetaan myös tieto siitä, miten taulujen tiedot liittyvät toisiinsa, ja sitä kutsutaan relaatioksi. Relaatio luodaan niin, että esimerkiksi ostokset-tauluun merkitään asiakkaan numero, joka on yksilöivä tieto, viiteavain, asiakkaat-taulussa, jossa asiakkaan tiedot löytyvät. Näin ollen asiakkaan nimi on vain yhdessä taulussa. (Sarja 2006.)

Tietokantasuunnittelu on interaktiivisen sivuston tärkein suunnitteluvaihe, sillä huolellisesti suunniteltu tietokanta on internet-sovelluksen perusta, joka helpottaa ohjelmointia ja luo hyvät jatkokehitysmahdollisuudet. Dynaamisen internet-sovelluksen tietokantasuunnittelun tavoitteena on luoda kaiken kattava tietokantarakenne sovelluksen pohjalle. Lisäksi hyvä tietokantarakenne on selkeä, joustava ja suorituskykyinen. Sovelluksen tietokantarakenteen suorituskyky tarkoittaa sovelluksen vastausaikaa. Mitä laajemmasta sovelluksesta on kyse, sitä tärkeämpää on, että tarvittava tieto saadaan tietokannasta mahdollisimman tehokkaasti. (Hovi ym. 2005, 20-21.)

5.1.2 Listaa.fi sivuston tietokantakaavio

Listaa.fi-sivuston tietokantasuunnittelun perustana, kuten aivan minkä tahansa verkkosivun, on sivuston toimintaperiaate. Listaa.fi-sivuston toiminnan tärkeimpänä elementtinä on ilmoitukset, joita käyttäjän tulee pystyä tallentamaan, katsomaan ja etsimään. Ilmoituksiin liittyy tietoa, jonka pohjalta tietokanta rakentuu.



Kuva 4: Työn tietokantarakenne

Ilmoitus on oma taulu, jonka tietoja ovat yksilöllinen ID, otsikko, kuvaus, hinta, editointikoodi, ilmoituksen jättöpäivämäärä ja ilmoituksen viimeisimmän muokkauksen päivämäärä. Lisäksi ilmoitusta koskee useampia viiteavaimia, joilla muodostetaan yhteys kahden taulun välille. Viiteavaimet ovat ilmoituksen tyyppi, kaupunki, kategoria ja ilmoittaja. Ilmoittaja-taulussa tietokenttiä ovat ID, nimi, puhelinnumero ja sähköpostiosoite.

Kaupunki-taululla on viiteavaimena lääni, jossa kaupunki sijaitsee. Näin ollen ilmoitukset saadaan listattua helposti alueiden ja kaupunkien mukaan. Kaupungilla ja läänillä on lisäksi yhtenä kenttänä nimi sekä yksilöllinen ID.

Lisäksi ilmoitus toimii viiteavaimena Ilmoituskuvan tauluun, jonka tarkoituksena on listata tiedostoon tallennetut ilmoituksen kuvat. Kuvalla on lisäksi image-kenttä, jossa on tieto kuvan tiedostonimestä.

Listaa.fi-sivuston tietokannan taulurakenne on yksinkertainen. Siinä on seitsemän taulua, joihin tietoa tallennetaan ja haetaan. Tietokanta on pyritty rakentamaan nopeaksi ja tehokkaaksi. Tietoa on helposti saatavilla yksinkertaisilla tietokantahauilla.

5.1.3 Tietokannan rakentaminen

Listaa.fi -sivuston tietokanta on rakennettu käyttäen PhpMyAdminia. Se on MySQL-tietokannan hallintatyökalu, jota voi käyttää graafisesti selaimen kautta tuntematta

varsinaisia MySQL-komentoja. PhpMyAdmin on PHP-kielellä rakennettu avoin sovellustyökalu, jolla saa helposti pystytettyä laajan relaatiotietokannan. (PhpMyAdmin Devel Team, 2012.)

5.2 Käyttöliittymä ja käytettävyys

Jokaisessa tietojärjestelmässä tulee olla käyttöliittymä, joka on näkymä käyttäjän ja sovelluksen välissä. Sen avulla käyttäjä kommunikoi sovelluksen kanssa. Käytettävyys on yleisesti ottaen Internet-sivuston pääkohta, koska monien sivujen pääasiallisena tarkoituksena on esitellä tai kaupata jotain Internetin käyttäjille. (Järvinen, P. 2001, 355.)

Käyttöliittymän tärkein ominaisuus on sen käytettävyys, jolla tarkoitetaan laitteen tai ohjelman käytön helppoutta (Järvinen 2012, 354). Käytettävyys määritellään web-suunnittelussa tuotteen tai palvelun laatuominaisuudeksi, jolla kuvataan, kuinka helposti ja tehokkaasti sovellusta pystyy käyttämään (Nielsen 2003).

ISO 9241-11 -standardin mukaan käytettävyys on mittari, jolla mitataan käytön tehokkuutta, mielekkyyttä ja tuottavuutta. Jokaisella kolmella termillä on myös oma määrittelynsä. Tuottavuus tarkoittaa tekemisen täydellisyyttä ja virheettömyyttä. Mielekkyys puolestaan kertoo, kuinka sujuvaksi käyttäjä toteaa käyttöliittymän käytön ja tehokkuus taas mittaa tuotteen käytössä vaadittavia resursseja. (ISO/IEC, 1998.)

5.3 Käyttöliittymä listaa.fi:ssä

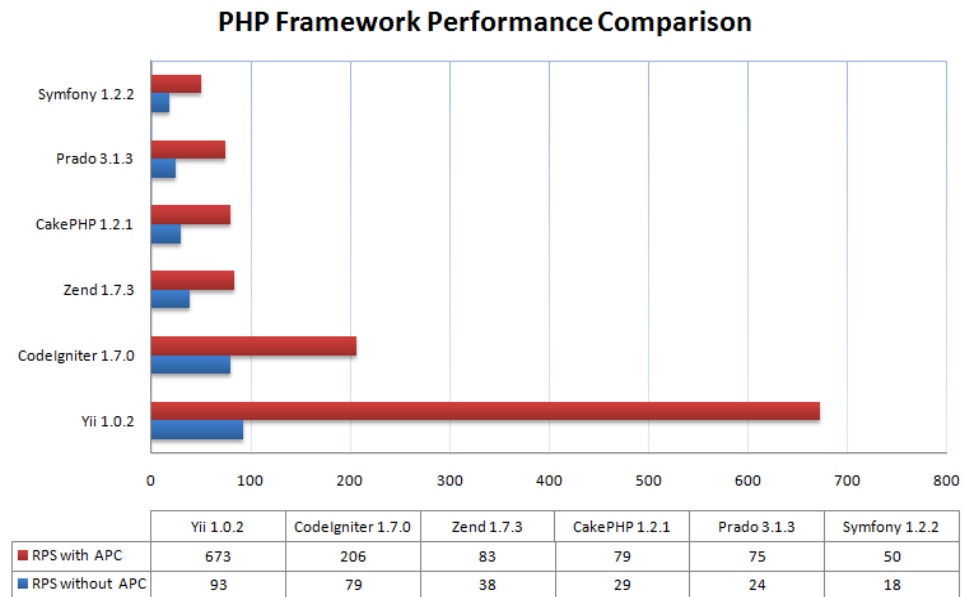
Toimeksiantaja Agilo Partners suunnitteli Listaa.-fi sivun ulkoasun ja eri näkymät. Tyyllisesti sivustossa tavoiteltiin yksinkertaista ilmoitusalustaa, jossa käytettävyys on ensisijaisen tärkeää, ja ulkoasu toissijainen.

Yhteiselle sivujen yleisilmeelle annettiin rauhoittava sininen taustaväri, ja paljon ilmaa elementtien väliin. Sivuilla ei ole pieniä yksityiskohtia, vaan kaikki elementit ovat isoja ja näkyviä. Sivun asettelu noudattaa Internetin peruskäytäntöjä; navigointi on ylälaidassa ja lisätietokentät sekä tekijänoikeusmerkintä ovat sivun alalaidassa. Listaa.fi-sivustossa on ainaoastan kaksi päänavigointia: selaa ilmoituksia ja jätä ilmoitus.

Jokainen sivu mukailee toistaan listaa.fi-sivustossa, ja se luo käyttäjälle jatkuvuuden tunnetta. Käyttäjän on helppo oppia käyttämään sivua, kun elementit pysyvät paikoillaan, eivätkä liiku. Ilmoitusten selaus ja ilmoituksen jättö -painikkeet ovat aina samassa kohdassa. Elementtien sijoittelu näiden sivujen sisällä on loogista ja toistuvaa, joten käyttäjän on helppo navigoida sivustolla.

5.4 Toteutus ohjelmistokehityksen avulla

Listaa-fi-sivusto toteutettiin hyödyntäen Yii-ohjelmistokehystä. Käytettävää ohjelmistokehystä valittaessa otettiin huomioon PHP-kielen käyttö, avoin lähdekoodi sekä ohjelmistokehityksen suosio. Yii-kehityksen saa sen kotisivuilta, josta ladataan paketti, joka sisältää kaikki ohjelmistokehitykseen kuuluvat luokat ja toiminnot.



Kuva 5: Ohjelmistokehitysten suoritusnopeudet (Lähde: yiiframework.com)

5.4.1 Teoria Yii:stä

Yii on olio-ohjelmointia hyödyntävä, korkeasuorituskykyinen, PHP 5 -kielen päälle kirjoitettu web-ohjelmistokehys. Sen tarkoituksena on helpottaa suurien web-sovellusten kehitystä ja ylläpitoa. Lisäksi Yii antaa oivan mahdollisuuden luoda sovelluksista helposti laajennettavia ja tehokkaita. Yii-ohjelmistokehys muotoutuu MVC-mallin (Model-View-Controller) ympärille, jolloin kehittäjän koodin määrä ja toisto minimoituu. Ohjelmointi on tehokasta ja työprosessi nopeutuu. (Winesett 2010, 8.)

__get() method

<code>public mixed __get(string \$name)</code>		
\$name	string	the property name or event name
{return}	mixed	the property value, event handlers attached to the event, or the named behavior

Source Code: [framework/base/CComponent.php#108](#) (show)

Returns a property value, an event handler list or a behavior based on its name. Do not call this method. This is a PHP magic method that we override to allow using the following syntax to read a property or obtain event handlers:

```
$value=$component->propertyName;
$handlers=$component->eventName;
```

See Also

- [__set](#)

__isset() method

<code>public boolean __isset(string \$name)</code>		
\$name	string	the property name or the event name
{return}	boolean	

Source Code: [framework/base/CComponent.php#185](#) (show)

Checks if a property value is null. Do not call this method. This is a PHP magic method that we override to allow using `isset()` to detect if a component property is set or not.

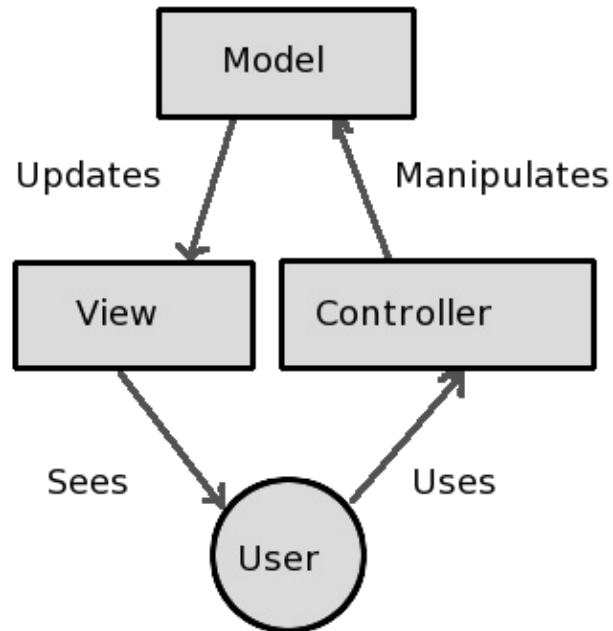
Kuva 6: Esimerkki Yii:n dokumentaatiosta sen kotisivuilla

5.4.2 MVC-malli

MVC-malli (Model-View-Controller) on ohjelmistoarkkitehtuurin tyyli, jossa sovelluksen koodi jaetaan kolmeen osaan: malliin, näkymään ja kontrolleriin. Malli (Model) on luokka, joka tyypillisesti käsittelee tietokannan tietojen ominaisuuksia, joita voidaan antaa sääntöinä malliluokalle. Malliluokka sisältää dataa, joka on tullut tietokannasta tai on sovelluksen käyttäjän syöttämää tietoa. Näkymä (View) on mallin tiedoista koostuva käyttöliittymä, joka näytetään käyttäjälle. Yii:ssä näkymä on tavallisesti PHP-skripti, joka sisältää ulkoasun HTML-elementit, mutta näkymä voi myös sisältää yksinkertaisia PHP-lausuntoja. Kontrolleri (Controller) on pääosa, joka toimii näkymän ja mallin välillä. Se ottaa tietoa käyttäjältä ja siirtää sitä mallin luokkaan, ja päivittää näkymää toimintojen mukaan. Kun kontrolleri suoritetaan, se kutsuu tarvittavan kontrollerin metodin (Action), joka on vuorovaikutuksessa tarvittavan mallin kanssa ja renderöi asianmukaisen näkymän. (Winesett 2010, 10.)

Tyypillinen palvelinpyyntö MVC-arkkitehtuurin internet-sovelluksessa on seuraavanlainen. Selain lähettää pyynnön palvelimen MVC-arkkitehtuurisovellukselle. Kontrolleri käsittelee

pyynnön ja on vuorovaikutuksessa mallin kanssa. Kontrolleri kutsuu näkymää, joka renderöi oikeanlaisen HTML-näkymän käyttäjän selaimeen. (Winesett 2010, 11.)



Kuva 7: Model-Controller-View -arkkitehtuurin rakenne

5.4.3 Hyödyt ohjelmistokehuksesta

PHP-ohjelmistokehysten käyttö jopa pienessä ohjelmistoprojektissa on suotavaa, sillä ohjelmistokehysten käytöstä on vain hyötyä. Sen oppiminen vie hetken aikaa, mutta kun on kerran opetellut käyttämään ohjelmistokehystä, on sen käyttäminen jatkossa todella tehokasta. Tämän opinnäytetyön tekijä opetteli Yii-ohjelmistokehysten käyttämistä Yii:n sivuilta löytyvän tutoriaalin turvin. Oppiminen tuntui aluksi todella vaikealta, mutta nopeasti sen rakenteeseen tottui.

Ohjelmistokehysten suurin hyöty on ohjelmoinnin nopeus ja tehokkuus. Kehys sisältää monet tarvittavat koodit, joten koodin toiston määrä vähenee huomattavasti. Ohjelmistokehys hoitaa muun muassa tietokantakyselyn muodot, jolloin ohjelmoija voi keskittyä vain tehokkaiden kyselyiden rakentamiseen. Ohjelmistokehys myös huolehtii, että sovelluksen arkkitehtuurinen rakenne on järjestyksessä, mikä myös nopeuttaa ohjelmointia. (Zahasman 2007.)

Tärkeä ohjelmistokehityksen hyöty on tietoturva, sillä ne ovat rakennettu turvallisiksi. Ohjelmistokehitys huolehtii suuren osan perinteisistä tietoturva-ongelmista. Näin ollen ohjelmistokehitystä käyttäen voi rakentaa käyttäjien täytettäviä lomakkeita ilman omia tietoturvasuoritusvarmistuksia. (Zahasman, 2007.)

6 Työn lopputulos

Työn lopputuloksena on toimiva ilmoitussivusto ilman suurempia hienouksia. Sivustolla käyttäjä voi joko jättää ilmoituksen täyttämällä kaavakkeen, tai selaamalla ilmoituksia haluamallaan tavalla. Sivusto ei vaadi kirjautumista eikä se sisällä käyttöehtojen ja palvelutietojen lisäksi muita sivuja kuin ilmoitukseen liittyviä sivuja.

6.1 Etusivu

Palvelun etusivulla näkyy oletuksena ilmoituksen jättö -lomake sekä näyteikkuna, jossa on nostona joko viisi uusinta tai eniten näytökertoja kerännyttä ilmoitusta. Etusivun oikealla laidalla on mainostilaa, jotta Agilo Partners saa kerättyä rahaa toimintaansa sivostolla. Etusivulla voi AJAX-kutsun avulla vaihtaa ilmoitusten selaus -lomakkeen ilmoituksen jättö -lomakkeen tilalle.

The screenshot shows the Listaa.fi homepage. At the top left, there's a search form with tabs for 'Selaa ilmoituksia' and 'Jätä ilmoituksia'. The search form includes fields for 'Ilmoituksen tyyppi' (with radio buttons for 'Myydään', 'Ostetaan', 'Vaihdetaan', 'Tarjotaan vuokralle', 'Halutaan vuokrata'), 'Osasto', 'Maakunta', 'Kunta', and 'Kirjoita hakusana'. Below the search form are tabs for 'Näyteikkuna', 'Uusimmat', and 'Katsotuimmat'. The 'Katsotuimmat' tab is active, showing a grid of five listing cards. The first card is for a game ('Peli') priced at 5,50 €. The other four cards are marked as unavailable with a large red 'X' and show prices: 'testiilmo' (1 300,00 €), 'Kello' (12,90 €), 'Maatila, 500 hehtaaria. 2 kanalaa + asuintalo.' (99 999 999,99 €), and 'testi' (100,00 €). On the right side, there's a vertical banner with the 'if...' logo and text: 'Osta, myy, vaihda tai vuokraa. Listaa.fi on Suomen kätevin ilmoitussivusto. Ilmoita ilmaiseksi ja ilman rekisteröitymistä nyt ja aina.'

Kuva 8: Listaa.fi -etusivu

listaa.fi

Selaa ilmoituksia **Jätä ilmoituksia**

Otsikko *

Kuvaus *

Hinta * €

Ilmoituksen tyyppi * Valitse tyyppi ▾

Lääni Valitse lääni ▾

Kaupunki * ▾

Yläkategoria Valitse yläkategoria ▾

Alakategoria * ▾

Kuvat: Choose File No file chosen

Nimi *

Puhelinnumero

Sähköposti *

Luo ilmoitus

@ listaa.fi 2011 | Tietoa palvelusta | Käyttöehdot | Selaa ilmoituksia | Jätä ilmoitus

Kuva 9: Ilmoituksen jättö -sivu

6.2 Ilmoitusten selaus

Ilmoitusten selaus -sivulla ilmoituksia voi selata käyttäjän mieltymysten mukaan. Ilmoituksia voi listata läänin, kaupungin, ilmoituksen tyyppin, kategorian tai sanahaun mukaan yhdessä tai erikseen. Ilmoituksia listautuu halvimman tai uusimman ilmoituksen mukaan.

Selaa ilmoituksia **Jätä ilmoituksia**

- Valitse osasto - ▾ - Valitse lääni - ▾ ▾

Myydään Ostetaan Vaihdetaan Tarjotaan vuokralle Halutaan vuokrata

Näytä ensin: Halvimmat | Uusimmat

Kuva 10: Ilmoituksen hakukriteerit

6.3 Ilmoitussivu

Ilmoitussivulla on ilmoituksen perustiedot, eli otsikko, ilmoitustyyppi, ilmoituksen sijainti, hinta sekä kuvia ilmoituksesta. Ilmoitukselle voi antaa useita kuvia, mutta ensimmäinen kuva


on oletuskuva, joka näkyy listauksissa. Ilmoituksen kuvat muuttuvat AJAX-latauksella kuva-alueelle, kun pientä kuvaa klikkaa. Ilmoituksen tietojen alapuolella on ilmoittajan nimi ja yhteystiedot.


listaa.fi


Muumiastiasto

Myydään | Tyyppi: Keräily | Ilmoitus jätetty: 20.11.2011 klo. 20:02

Hakutulos | Seuraava







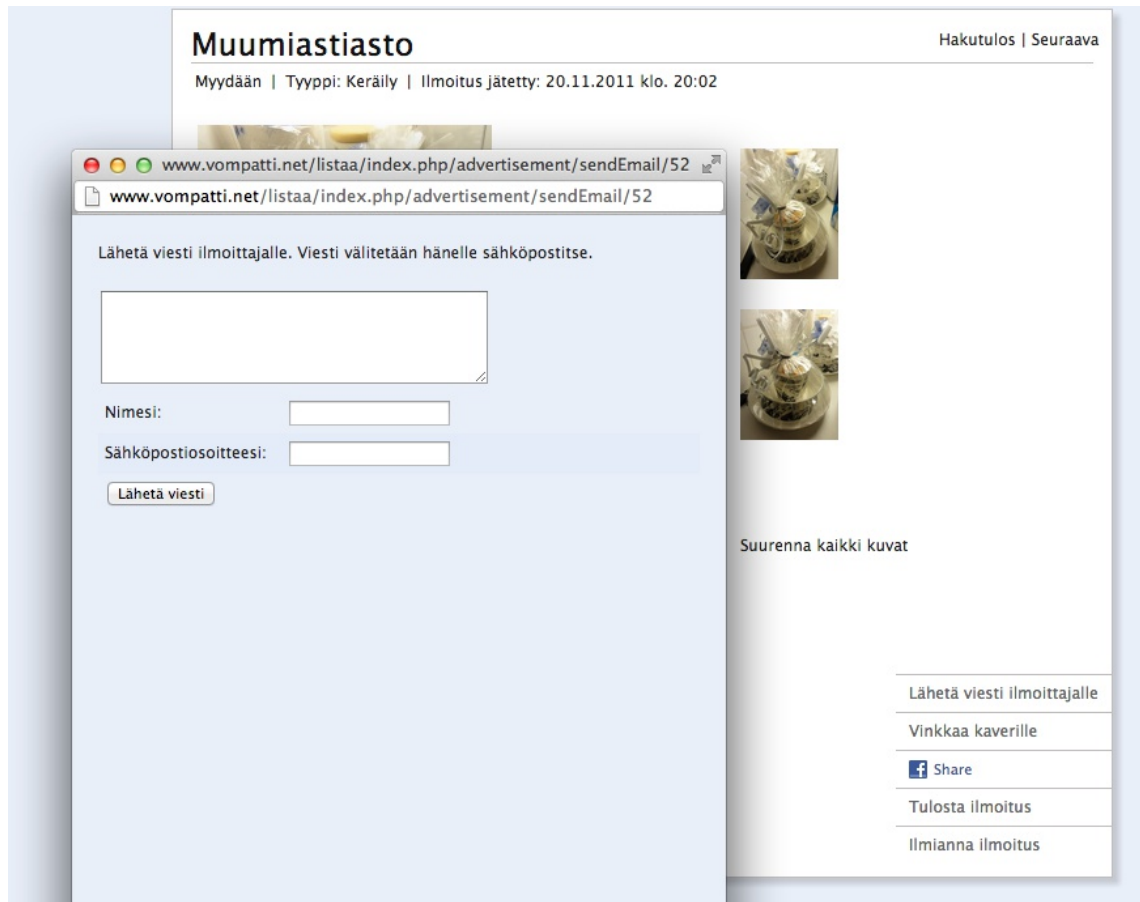
Suurena kaikki kuvat

Muumi seikkailu -astiasto

Hinta:	25,50 €				
Sijainti:	Espoo, Uusimaa			Lähetä viesti ilmoittajalle	
Ilmoittaja:	Tiina Testaaja			Vinkkaa kaverille	
Puhelin:	040112233			Share	
Sähköposti:	hanna.taimen@gmail.com			Tulosta ilmoitus	
				Ilmianna ilmoitus	

Kuva 11: Ilmoitussivu

Lisäksi ilmoitukseen liittyy tiettyjä toimintoja; ilmoituksen voi ilmiantaa, sen voi jakaa Facebookissa tai kaverille sähköpostilla tai ilmoittajaan voi ottaa suoraan yhteyttä lomakkeella. Näistä Facebook-jako painike on Facebookin oma liitännäinen (eng. plug-in), mutta muut on itse rakennettuja. Lähetettävät lomakkeet aukeavat popup-ikkunaan, jolloin käyttäjän ei tarvitse poistua ilmoitussivulta. Ilmoituksia voi selata eteen- ja takaisinpäin valitsemien hakukriteerien mukaisesti.



Kuva 12: Viestin lähetyksen ilmoittajalle -toiminto

6.4 Yhteenveto

Tämän opinnäytetyön tavoitteena oli toteuttaa Agilo Partners -yrityksen toiveiden ja ulkoasuun suunnitelmien mukainen ilmoitussivusto internetiin. Työ lähti ohjelmointikielen ja ohjelmistokehityksen valitsemisesta ja niihin tutustumisesta. Ohjelmointikieleksi valikoitui PHP-kieli ja tietokantana käytettiin MySQL-tietokantaa, ja ohjelmistokehityksenä toimii Yii-kehys. Sivuston vaatimusmäärittely luotiin yhdessä toimeksiantajayrityksen kanssa, ja ulkoasu tuli toimeksiantajalta. Ulkoasu oli nykyaikainen, ykinkertainen ja ilmava, joten siihen ei ollut puuttumista. Sivuston lähtökohtana on käyttäjien jättämät ilmoitukset, eikä sen ympärille kaivattu muuta kuin toimiva alusta niiden selaamiseen ja jättämiseen.

Yii-ohjelmistokehityksen opettelu oli helppoa, sillä heidän tarjoamillaan kotisivuilla on laaja dokumentaatio listatuista ominaisuuksista, metodeista ja luokista, joita Yii sisältää. Lisäksi sivuilla on toimiva tutoriaali, jota seuraamalla harjoitustyönä luodaan blogipohja. Tutoriaalin tarkoituksena on opettaa Yii:n käyttö, ja esittää, kuinka helppoa ohjelman tai sovelluksen luominen sen päälle on. Yii:n sivusto sisältää kattavan dokumentaation sen toiminnoista, ja lisäksi sivustolla on keskustelufoorumi kehittäjien ongelmien ratkomiseksi.

Tämän opinnäytetyön teoria keskittyi tukemaan pienen ohjelmistokehitysprojektin toteutusta. Teoriaosuudessa avattiin yleisimpiä web-ohjelmoinnin termejä, nykyaikaisia tekniikoita sekä kuvattiin pienen projektin toimintamalli, joka koostuu määrittelystä, suunnittelusta, toteutuksesta ja käyttöönotosta. Toteutusvaiheessa huomio kiinnitettiin Yii-ohjelmistokehityksen käyttöön projektin alustana ja MVC-mallin etuihin verrattuna siihen, että projekti ohjelmoidaan alusta alkaen itse, tyhjän päälle.

Opinnäytetyön toteutus oli alun Yii:n opetteluun jälkeen helppoa ja nopeaa. Yii-ohjelmistokehitys toimi tämän kokoisen sivuston toteutuksen alustana erinomaisesti, ja soveltuu varmasti monenlaisen ja -kokoisen web-ohjelmoinnin projektin toteutukseen. Yii:n etuna on sen tehokkuus ohjelmoijan näkökulmasta. Yii sisältää hyödyllistä koodia ja luokkia, joita voi käyttää suoraan omaan projektiin keskittyen ainoastaan olennaiseen. Yii hoitaa koodin selkeän rakenteen MVC-arkkitehtuurin turvin, tietokantayhteydet ja MySQL-kyselyt sekä tietoturva-aukot.

Opinnäytetyön tuloksena syntyi käyttökelpoinen internet-sivusto, jossa käyttäjät voivat jättää omia ilmoituksia myytävistä, vaihdettavista, ostettavista ja vuokrattavista tavaroista. Sivusto on sellaisenaan toimiva ja selkeä. Ennen käyttöönottoa se tulee vielä testata jokaiselle selaimella, ja todeta toimivaksi. Sovelluksen käyttöönotto ja testaus ovatkin kokonaan oma kokonaisuus, jatkokehitystä tämän työn loppuun saattamiseksi. Toimeksiantajayritys, Agilo Partners, oli tyytyväinen projektin lopputulokseen.

7 Loppusanat

Opinnäytetyön aihe oli mielenkiintoinen ja ammatillisen kasvun kannalta erinomainen. Yrityselämässä jokainen projekti rakennetaan jonkun ohjelmistokehityksen päälle, koska on turhaa keksiä pyörää uudestaan. Ohjelmistokehityksen käyttö nopeuttaa ohjelmistoprojektia merkittävästi. Yhden MVC-arkkitehtuuria hyödyntävän ohjelmistokehityksen osaaminen takaa nopean oppimisen muiden ohjelmistokehitysten käyttöön, joten ammatillisen kasvun kannalta tästä opinnäytetyöstä on ollut paljon hyötyä.

Opinnäytetyön toiminnallinen työ sujui nopeasti, lähes puolessa vuodessa. Yii:n opettelu oli ajallisesti pisin vaihe, mutta sen jälkeen toteutus sujui nopeasti, sillä määrittely ja suunnittelu tehtiin huolellisesti yhdessä toimeksiantajan kanssa. Ammatillisen kasvun kannalta oli hyvä huomata, että määrittely ja suunnittelu ovat tärkeä vaihe ennen toteutuksen aloittamista, joita ei missään nimessä saa aliarvioida, vaan niihin on käytettävä ajallisesti lähes yhtä paljon aikaa, kuin toteutukseen. Suunnitteluvaiheessa virheet ovat paljon helpompi korjata kuin työn toteutusvaiheessa.

Oli tärkeää ottaa yksin vastuu projektista ja sen etenemisestä sekä kommunikoinnista toimeksiantajan kanssa. Yrity maailmassa on tärkeää osata ottaa vastuuta ja usein pienet ohjelmistokehitysprojektit hoidetaan yhden tai kahden ihmisen voimin. Silloin on tärkeää osata päästä ymmärrykseen asiakkaan tarpeista sekä määrittellä ne huolellisesti ja pitää kiinni sovituksista. Opinnäytetyön lopputuloksena syntynyt listaa.fi -sivusto vastasi asiakkaan tarpeita, ja asiakas oli tyytyväinen tuotokseen.

Lähteet

Ekonoja, Antti & Lahtonen, Tommi & Mäntylä Jukka. Jyväskylän yliopiston IT-tiedekunta ja avoin yliopisto. Luennot - AJAX. 2010.
Viitattu 21.10.2012.

<http://appro.mit.jyu.fi/sovellukset/luennot/luento13/>

jQuery Foundation. 2012.

Viitattu 27.10.2012.

<http://jquery.com>

Maurer, Donna. 2006. Usability for Rich Internet Applications. Digital Web Magazine.

Viitattu 14.3.2011.

http://www.digital-web.com/articles/usability_for_rich_internet_applications/

Nielsen, Jakob. Usability 101: Introduction to usability, 25. elokuuta 2003. ISSN 1548-5552.

Viitattu 30.11.2012.

<http://www.useit.com/alertbox/20030825.html>

Php My Admin Devel Team. 2003-2012.

Viitattu 30.11.2012

http://www.phpmyadmin.net/home_page/index.php

Sarja, Jari. 2006. Relaatiotietokanta.

Viitattu 30.11.2012.

<http://www.verkkopedagogi.net/vanhat/fi/sisalto/materiaalit/access2003/luku0375c6.html?C:D=419702&selres=419702>

Zahasman, Tengku. 2007. Why you should use a Web Application Framework.

Viitattu 8.11.2012

<http://web2.0entrepreneur.com/7/why-you-should-use-a-web-application-framework.html>

Haikala, Ilkka & Märijärvi, Jukka. 2004. Ohjelmistotuotanto. Talentum Media Oy. Helsinki.

Heinisuo, Rami & Rauta, Ilkka. 2007. PHP ja MySQL - Tietokantapohjaiset verkkopalvelut. Talentum Media Oy. Helsinki.

Hovi, Ari & Huotari, Jouni & Lahdenmäki, Tapio. Tietokantojen suunnittelu & indeksointi. 2005. WS Bookwell. Porvoo.

ISO/IEC. ISO 9241-11: Ergonomic requirements for office work with visual display terminals VDTs - Part 11: Guidance on usability. Standard, International Standardization Organization, 19. lokakuuta 1998.

Keogh, James. 2005. JavaScript Demystified. McGraw-Hill Professional Publishing.

Mohapatra, Pratap. 2010. Software Engineering. New Age International Publishers. Delhi.

Peltomäki, Juha. 2001. JavaScript. Docendo Finland Oy. Jyväskylä.

Rantala, Ari. 2002. PHP - Web-ohjelmoinnin peruskirja. WS Bookwell. Porvoo.

Ruohonen, Mika. 2002. Tietoturva. WS Bookwell. Porvoo.

Vilka, Hanna & Airaksinen, Tiina. 2003. Toiminnallinen opinnäytetyö. Gummerus kirjapaino Oy. Jyväskylä.

Winesett, Jeffrey. 2010. Agile Web Application Development with Yii 1.1 and PHP5. Packt Publishing Ltd. Birmingham.

Kuvat

Kuva 1: Internetin toiminta. Selain tekee pyynnön, jonka palvelin palauttaa käyttäjän selaimelle.....	10
Kuva 2: HTML-koodin rakenne.....	10
Kuva 3: Esimerkki vesiputousmallista.....	13
Kuva 4: Työn tietokantarakenne.....	16
Kuva 5: Ohjelmistokehysten suoritusnopeudet (Lähde: yiiframework.com).....	18
Kuva 6: Esimerkki Yii:n dokumentaatiosta sen kotisivuilla.....	19
Kuva 7: Model-Controller-View -arkkitehtuurin rakenne.....	20
Kuva 8: Listaa.fi -etusivu.....	21
Kuva 9: Ilmoituksen jättö -sivu.....	22
Kuva 10: Ilmoituksen hakukriteerit.....	22
Kuva 11: Ilmoitussivu.....	23
Kuva 12: Viestin lähetys ilmoittajalle -toiminto.....	24
Kuva 13: Ilmoitukseen liittyvät toiminnot.....	24