

SAIMAAN AMMATTIKORKEAKOULU
Tekniikka Lappeenranta
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka

Jussi Loiri

HYVÄN OLON PÄIVÄKIRJAN YLLÄPITOSIVUT

Opinnäytetyö 2010

TIIVISTELMÄ

Jussi Loiri

Hyvän Olon Päiväkirjan ylläpitosivut, 49 sivua, 2 liitettä

Saimaan ammattikorkeakoulu, Lappeenranta

Tekniikka, tietotekniikan koulutusohjelma

Ohjelmistotekniikka

Opinnäytetyö, 2010

Ohjaajat: DI Jouni Könönen, Saimaan ammattikorkeakoulu, Fysioterapeutti Milla Ranta, Saimaan ammattikorkeakoulu

Tässä opinnäytetyössä on luotu ylläpitosivut Hyvän Olon Päiväkirja -web-sovellukselle. Hyvän Olon Päiväkirja on Saimaan Ammattikorkeakoulun henkilökunnan käytössä oleva elämäntapojen tarkkailuun ja kohentamiseen tarkoitettu sivusto. Työ toteutettiin web-ohjelmoinnin dynaamisia tekniikoita käyttäen. Ylläpitosivujen tarkoituksena on Hyvän Olon Päiväkirjan pääsivujen sisällön helppo hallinta. Tämä tarkoittaa, että ylläpitosivujen käyttöliittymän tulee olla selkeä ja sellainen, että lähes kuka tahansa voi toimia sivujen ylläpitäjänä.

PHP on yksinkertaistettujen ominaisuuksiensa vuoksi kätevä tekniikka web-sovellusten nopeaan kehitykseen. Kielen ongelmana kuitenkin on toteutustapoihin liittyvien standardien ja ohjeiden puuttuminen. Monet toteutustavat perustuvat vieläkin aikaan, jolloin PHP:llä kirjoitettiin lyhyitä komentosarjoja isompien sivukokonaisuuksien sijaan.

Työssä pohditaan, kuinka PHP-pohjaiset sivut olisi järkevintä toteuttaa. Työn tarkoituksena ei kuitenkaan ole kertoa, kuinka PHP-sivut tulisi aina toteuttaa, vaan työssä käytetyt ratkaisut perustuvat opinnäytetyön aiheen asettamiin vaatimuksiin. Työ itsessään tarjoaa yhden tavan dynaamisen web-sovelluksen toteuttamiseen. Kaikki työssä käytetyt tekniikat ja työkalut ovat ilmaisia.

Työ aloitetaan esittelemällä Hyvän Olon Päiväkirja, sovellus johon ylläpitosivut on toteutettu, sekä asiakas, Saimaan ammattikorkeakoulu. Tämän jälkeen työssä esitellään kaikki työn toteutukseen käytetyt tekniikat, työn kulku vaiheelta ja valmiit ylläpitosivut sekä niiden toteutuksessa käytetyt ratkaisut. Lopuksi pohditaan kehitysmahdollisuuksia ja vedetään yhteen työn tulokset.

Asiasanat: Dynaaminen web-sovellus, PHP, CSS, HTML, MySQL

ABSTRACT

Jussi Loiri

Content management system of Hyvän Olon Päiväkirja, 49 pages, 2 appendices

Saimaa University of Applied Sciences, Lappeenranta

Technology, Degree Programme in Information Technology

Software Engineering

Bachelor's Thesis, 2010

Instructors: M.Sc. Jouni Könönen, Saimaa University of Applied Sciences, Physiotherapist Milla Ranta, Saimaa University of Applied Sciences

The goal of this thesis was to create a content management system for Hyvän Olon Päiväkirja web application. Hyvän Olon Päiväkirja (The diary of well-being) is a web application which is used by the staff of Saimaa University of Applied Sciences. It is created for observing and improving of lifestyle by guiding towards a healthier way of living. The work was implemented by using the dynamic web development techniques. The purpose of the content management system is to manage the content of Hyvän Olon Päiväkirja effortlessly. This means that the system should be so easy to use that anyone can act as an administrator of the application.

Because of its simplified properties, PHP is a handy technique for agile implementation of web applications. The problem of the language is the lack of standards and guidelines used in the implementation. Often the ways of implementation are based on the times when PHP was mere a language designed for writing of short scripts instead of larger web applications.

The document discusses which would be the most sensible way of implementing PHP-based web applications. However, the purpose is not to tell how PHP-based web applications should always be done. All the solutions used in this work are based on the demands of the topic itself. The work only offers one way of implementing the dynamic web application. All the techniques and tools used in this work are free to use.

The study begins with the introductions of the client and the application in which the content management system is implemented. After the introductions the study focuses on the work itself. It introduces the techniques used in this work, all the stages of the work one by one, the finished product and the solutions used in its implementation. Finally, it provides discussion about the future development possibilities of the finished product and closes with a summary of the entire work.

Keywords: Dynamic web application, PHP, CSS, HTML, MySQL

SISÄLTÖ

1	JOHDANTO.....	8
2	HYVÄN OOLON PÄIVÄKIRJA	9
3	ASIAKKAAN TOIMINNAN KUVAUS	10
4	TYÖSSÄ KÄYTETYT MALLIT, MENETELMÄT, TEKNIIKAT JA TYÖVÄLINEET	12
4.1	Vesiputousmalli.....	12
4.2	Dynaaminen web-dokumentti	14
4.3	PHP	15
4.4	MySQL ja SQL.....	16
4.5	Hypertext Markup Language.....	17
4.6	Cascading Style Sheets.....	17
4.7	JavaScript.....	18
4.8	WampServer 2.....	19
4.9	SciTE-tekstieditori.....	19
5	PHP TOTEUTUSVÄLINEENÄ.....	20
5.1	PHP:n valinnan perusteet	20
5.2	Aikaisemmin kohtaamani ongelmat	21
5.3	PHP:n ja asiakaspuolen tekniikoiden yhdistäminen	22
6	SUUNNITTELUN LÄHTÖKOHDAT	22
7	HYVÄN OOLON PÄIVÄKIRJAN YLLÄPITOSIVUJEN TOTEUTUKSEN VAIHEET	23
7.1	Esitutkimus	23
7.2	Suunnittelu.....	24
7.3	Toteutus.....	25
7.4	Testaus.....	30
7.5	Käyttöönotto.....	31
8	YLLÄPITOSIVUJEN ESITTELY	31
8.1	Sivurakenne.....	32
8.2	Sivun toimintaperiaate	34
8.3	Sivujen sisältö.....	36
9	KEHITTÄMISMAHDOLLISUUDET	43
10	POHDINTA JA YHTEENVETO	44
	KUVAT.....	47
	TAULUKOT.....	47
	LÄHTEET	48

LIITTEET

Liite 1 Lähdekoodiesimerkki

Liite 2 Kuvankaappauksia ylläpitosivuilta

TERMIT JA LYHENTEET

Ajax Asynchronous JavaScript and XML. Eri tekniikoista koostuva sovel-
luskehitystekniikka.

ASCII American Standard Code for Information Interchange. Standardoitu
koodi merkkien esittämiseksi (Tietotekniikan liitto ry 2003, 16).

ASP.net Active Server Pages. Microsoftin kehittämä .net-arkkitehtuuriin pe-
rustuva ohjelmointimenetelmä.

CSS Cascading Style Sheets. Tyylikieli.

Deklaraatio CSS-tyyliohjeissa käytettävä rakenne, joka koostuu ominaisuudesta
ja sitä vastaavasta arvosta (Korpela 1998).

Div-elementti

<div>. CSS-tyyliohjeissa ryhmittelyyn käytettävä elementti.

Elementti Dokumentin sisältöä ryhmittelevä rakenneos (Tietotekniikan liitto
ry 2003, 34).

Funktio Aliohjelma, joka kutsuttaessa suorittaa tietyn toiminnallisuuden.

HOP Hyvän Olon Päiväkirja.

HTML Hypertext Markup Language. Hypertekstin merkkäuskieli.

HTML-editori

Ohjelma, jolla luodaan WWW-sivuja

Java Ohjelmointikieli.

JavaScript Pääasiassa web-ohjelmoinnissa käytettävä komentosarjakieli.

Komentosarjakieli

Skriptikieli. Kieli jolla kirjoitetaan komentosarjoja. Esimerkiksi PHP.

Käyttöjärjestelmä

Ohjelmisto, joka ohjaa tai helpottaa sovellusohjelmien suoritusta
(Tietotekniikan liitto ry 2003, 99).

Käyttöliittymä

Toiminnot ja välineet, joilla käyttäjä on vuorovaikutuksessa ohjel-
man tai järjestelmän kanssa (Tietotekniikan liitto ry 2003, 99).

Linkki Tietojoukkoon kuten esimerkiksi tekstiin sisältyvä tai liittyvä tieto toisen tietojoukon sijainnista (Tietotekniikan liitto ry 2003, 111).

Luokka 1. Oliotekniikan malli, jonka ilmentymät ovat samankaltaisia olioita (Tietotekniikan liitto ry 2003, 118).

2. HTML:ssä elementteihin liitetty ominaisuus. CSS:ssä määritettä käytetään liittämään elementteihin erilaisia sääntöjä, vaikka elementit ovat keskenään samaa tyyppiä. (Korpela 1998.)

Lähdekoodi

Tekstimuodossa oleva ohjelmointikielellä kirjoitettu ohjelman toiminnallisuuden kuvaus.

Makro Sarja toimintoja, jotka on koottu yhdeksi kerrallaan suoritettavaksi kokonaisuudeksi (Tietotekniikan liitto ry 2003, 122).

Microsoft Ohjelmistoalan yritys.

Modulaarinen

Osiin jaettu kokonaisuus.

Moduuli Itsenäinen osa, jollaisista voidaan koota erilaisia kokonaisuuksia

Muuttuja Variable. Tieto, jonka arvo voi muuttua. Tunnus, jonka avulla viitataan tietokoneen muistiin tallennettuihin tietoihin. (Rantala 2002, 62, Hietanen 2004, 791)

MySQL Tietokannanhallintajärjestelmä.

Ohjelmointiympäristö

Ohjelmien tuottamista avustava ohjelmisto- ja laitekokonaisuus (Tietotekniikan liitto ry 2003, 151).

Olio Luokan ilmentymä. Tietojen ja palvelujen kokonaisuus (Hietanen 2004, 793).

Olio-ohjelmointi

Ohjelmointityyli, jossa käytetään olio-ohjelmointitekniikoita (Hietanen 2004, 792).

Parametri Kutsuvasta ohjelmasta aliohjelmaan välitettävä tieto.

PHP PHP: Hypertext Preprocessor. Kommentosarjakieli.

Rikastettu teksti

Rich Text Format, RTF. Tiedostomuoto, joka mahdollistaa tekstitiedostojen siirron erilaisten tekstinkäsittelyohjelmien ja erilaisten käyttöjärjestelmien välillä säilyttäen niiden muotoilun (Korpela 2000).

SQL Structured Query Language. Kyselykieli.

Tagi Tageja käytetään muun muassa HTML-sivujen muotoilussa. Elementillä pitää yleensä olla sekä aloitus- että lopetustagi, joiden väliin elementin sisältö tulee. <elementti>sisältö</elementti>.

Taulukko 1. Array. Rakenteinen tietotyyppi, jonka avulla yhteen muuttujaan voidaan tallentaa kerrallaan useita arvoja (Rantala 2002).
2. <table>. HTML-elementti.

Transaktio Joukko tapahtumia, jotka suoritetaan peräkkäin, aivan kuin kyseessä olisi yksi tapahtuma. Jos jokin tapahtuma epäonnistuu, koko transaktio perutaan. (Oulun seudun ammattikorkeakoulu.)

Vesiputousmalli

Projektityömalli

Web Katso WWW.

Web-selain Internet-verkon käyttäjän verkkoasemassa oleva selain, joka noutaa, tulkitsee ja esittää websivuja (Tietotekniikan liitto ry).

Web-sovellus

Selaimella käytettävä sovellus, joka jaetaan Internetin kautta (Tietotekniikan liitto ry).

WWW World Wide Web. Internet-verkossa toimiva tietojenkäsittelyympäristö, jonka toteutus perustuu ensisijaisesti HTML-kielisten sivujen siirtämiseen. (Tietotekniikan liitto ry 2003, 291.)

WWW-sivu Selaimella katseltava dokumentti.

XHTML eXtensible Hypertext Markup Language. Kieli, joka käyttää HTML:n sanastoa ja XML:n syntaksia (Boumphrey ym. 2000, 33).

Ylläpitosivut

Web-sovelluksen sisällön hallintaan tarkoitettut sivut.

1 JOHDANTO

Opinnäytetyön tarkoituksena on luoda ylläpitosivut Hyvän Olon Päiväkirja - sivustolle. Ylläpitosivut ovat nykyään olennainen osa WWW-sivuja. Niiden avulla on siirrytty staattisista verkkosivuista dynaamisiin web-sovelluksiin, joissa sisältö elää jatkuvasti. Ylläpitosivujen avulla ylläpitäjät voivat helposti hallita sivustoilla olevaa sisältöä ilman minkäänlaisia ohjelmointitaitoja tai tietämystä siitä, kuinka sivut ovat rakennettu.

Ylläpitosivujen pitää olla sellaiset, että niitä käytetään, koska mitä enemmän niitä käytetään, sitä enemmän sivut elävät. Käyttämättömät ylläpitosivut tarkoittavat usein sitä, että ne ovat liian vaikeakäyttöiset. Sovelluskehittäjien haasteena onkin luoda helppokäyttöiset, mutta samalla myös monipuoliset ja toimivat ylläpitosivut.

Myös Hyvän Olon Päiväkirjan ylläpitosivujen pitää täyttää edellä mainitut ylläpitosivuille asetetut vaatimukset. Sivujen avulla ylläpitäjät voivat hallita käyttäjätilejä sekä sivuston sisältöä. Asiakkaan toiveena on, että periaatteessa kuka tahansa voisi toimia sivuston ylläpitäjänä. Tämä tarkoittaa sitä, että kenen tahansa pitää osata käyttää ylläpitosivuja ilman suurempaa perehdyttämistä.

Valitsin opinnäytetyön aiheeksi Hyvän Olon Päiväkirjan ylläpitosivut, koska olin työskennellyt sivujen parissa jo puoli vuotta. Täten oli luonnollista, että tekisin myös sivuston ylläpitosivut. Myös sivuston käsittelemät aiheet, kuten liikunta ja terveelliset elämäntavat, kiinnostavat minua.

Nykyään kiinnitetään paljon huomiota muun muassa alkoholin ja tupakoinnin terveyshaittoihin, vaikka useampaa meistä uhkaavat epäterveellisen ruoan ja stressaavan elämän aiheuttamat vaivat ja sairaudet. On erittäin kummallista, ettei ihmisten epäterveellisiin ruokailutottumuksiin puututa näkyvämmiin, eikä ihmisten työ- tai opiskelupaineita helpoteta vaikka yhä useampi voi huonosti. Jatkuvan alkoholiverotuksesta ja tupakointilaista kiistelemisen sijaan pitäisi terveellisiä ruokailutapoja tukea, esimerkiksi laskemalla terveellisten elintarvikkeiden verotusta. Myös stressiä pitäisi pyrkiä ennalta ehkäisemään lisäämällä va-

paa-aikaa ja tukemalla mielenterveystyötä. Hyvän Olon Päiväkirjan tarkoituksena on ohjata ihmisiä kohti terveellisimpiä elämäntapoja ja tehdä se sormella osoittelematta.

2 HYVÄN OLON PÄIVÄKIRJA

Hyvän Olon Päiväkirja eli lyhyemmin HOP on Saimaan ammattikorkeakoulun henkilökunnan käytössä oleva elämäntapojen tarkkailuun ja kohentamiseen tarkoitettu sivusto. HOP:n kehitys alkoi keväällä 2009 Saimaan ammattikorkeakoulun fysioterapeutin Milla Rannan johdolla. Minut valittiin sivuston toteuttajaksi. Ranta ja minä suunnittelimme sivuston kesän 2009 aikana. Samana kesänä alkoi myös sivuston toteutus. Ensimmäinen versio otettiin käyttöön 2. joulukuuta 2009. Sivusto on ollut käytössä siitä lähtien, vaikka kehitettävääkin vielä löytyy. Kun tässä raportissa puhutaan pääsivuista, niillä viitataan juuri näihin sivuihin.

Hyvän Olon Päiväkirjassa käyttäjä syöttää ensimmäisenä henkilökohtaiset tietonsa kuten iän, painon, sukupuolen ja unentarpeen ja tallentaa ne sovelluksen tietokantaan. Näiden tietojen avulla ohjelma laskee muun muassa käyttäjän peruskulutuksen. Kun tiedot on syötetty ja tallennettu, voidaan sovelluksen käyttäminen aloittaa.

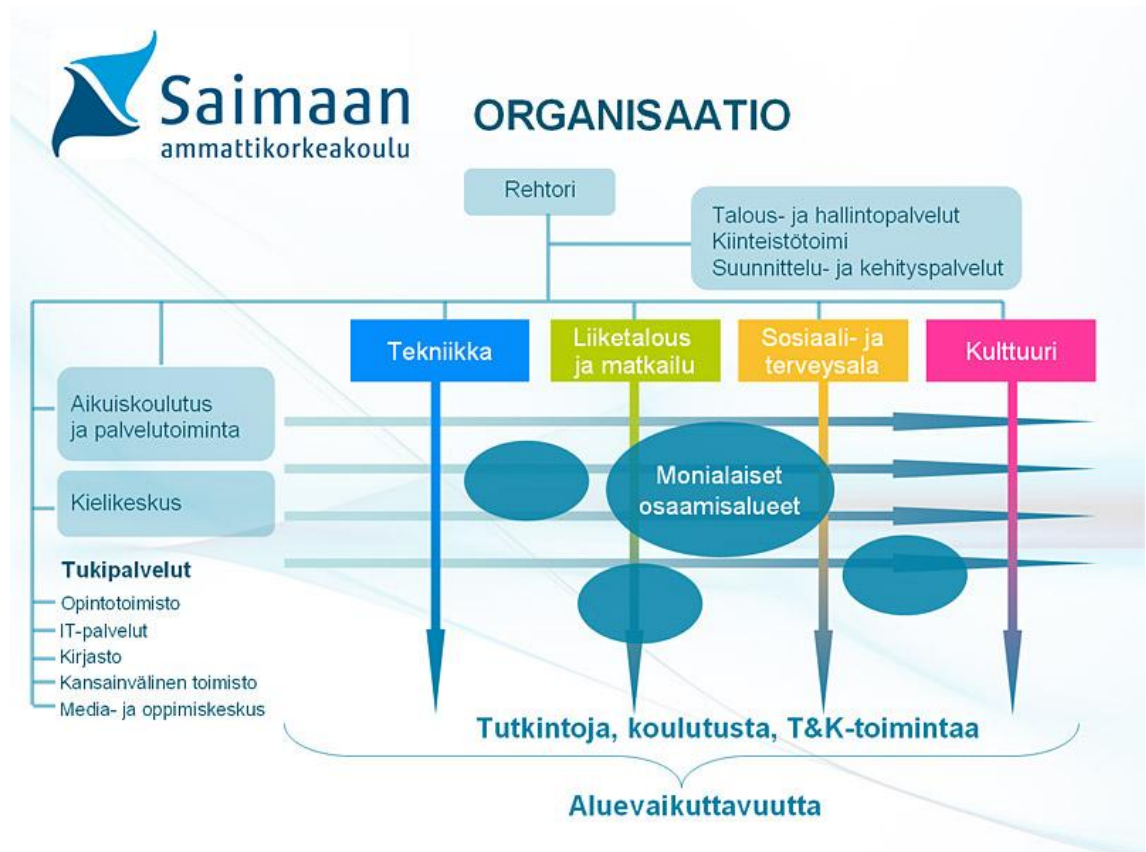
Sivuston avulla käyttäjä voi pitää sekä ravinto- että liikuntapäiväkirjaa. Ravintopäiväkirjassa merkitään ylös kaikki päivän aikana syödyt tuotteet. Sovellus laskee tuotteiden kokonaiskalorimäärän ja vertaa sitä käyttäjän peruskulutukseen, ilmoittaen onko kokonaiskalorimäärä yli vai alle peruskulutuksen. Liikuntapäiväkirjassa käyttäjä puolestaan merkkää ylös kaikki päivän aikana harrastamansa liikunnat. Sovellus laskee, kuinka paljon käyttäjä on liikkunut päivän aikana ja kuluttanut energiaa. Liikunnalla kulutettu energia lisätään päivän peruskulutukseen, jolloin käyttäjän on mahdollista syödä enemmän ylittääkseen peruskulutuksensa.

Sovelluksella voi pitää myös tupakointi- ja unipäiväkirjaa sekä tehdä erilaisia stressitestejä. Sovellus antaa käyttäjälle vinkkejä päiväkirjojen ja testien perusteella. Näiden lisäksi ohjelma sisältää paljon erilaista statistiikkaa ja tietoisuuksia.

3 ASIAKKAAN TOIMINNAN KUVAUS

Opinnäytetyön asiakas on Saimaan ammattikorkeakoulu. Asiakkaan edustajana toimii Saimaan ammattikorkeakoulun aikuiskoulutus ja palvelutoiminnassa työskentelevä fysioterapeutti Milla Ranta.

Saimaan ammattikorkeakoulu Oy eli lyhyemmin Saimaan AMK on Etelä-Karjalan koulutuskuntayhtymäkonserniin kuuluva oppilaitos. Saimaan AMK:ssa on viisi koulutusala: tekniikka, sosiaali- ja terveysala, liiketalous, matkailu- ja ravitsemispalvelut ja kulttuuri. Saimaan AMK:ssa on 18 ammattikorkeakoulututkintoon johtavaa koulutusohjelmaa, joista neljä on kokonaan englanninkielisiä. Edellä mainittujen lisäksi Saimaan AMK:ssa on neljä ylempään ammattikorkeakoulututkintoon johtavaa koulutusohjelmaa, joista yksi on kokonaan englanninkielinen. Saimaan ammattikorkeakoululla on neljä koulutuskampusta joista kaksi sijaitsee Imatralla ja kaksi Lappeenrannassa. Kansainvälisiä vaihtomahdollisuuksia on eri puolilla maailmaa. Kuvassa 3.1 on esitetty Saimaan AMK:n organisaatio. (Etelä-Karjalan koulutuskuntayhtymä; Saimaan ammattikorkeakoulu.)



Kuva 3.1 Saimaan ammattikorkeakoulun organisaatiokaavio (Saimaan ammattikorkeakoulu)

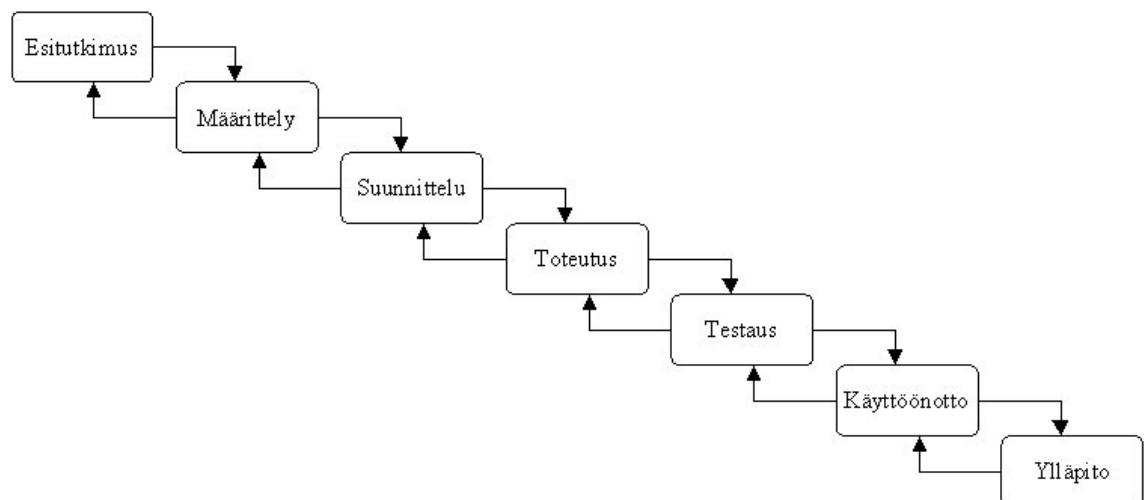
Saimaan AMK:n kokonaisopiskelijamäärä oli vuonna 2009 noin 3000. Opiskelijajakauma koulutusaloittain oli seuraavanlainen: opiskelijoista tekniikassa 39 %, sosiaali- ja terveysalalla 26 %, liiketaloudessa 21 %, matkailu- ja ravitsemispalveluissa 9 % ja kulttuurialalla 5 %. Saimaan AMK:n henkilöstömäärä oli noin 300 vuonna 2009. Talousarvio samana vuonna oli noin 24 miljoonaa euroa. (Saimaan ammattikorkeakoulu.)

4 TYÖSSÄ KÄYTETYT MALLIT, MENETELMÄT, TEKNIIKAT JA TYÖVÄLINEET

Projektien läpiviemiseen on tarjolla useita erilaisia malleja. Myös itse sovellusten suunnitteluun, toteutukseen ja testaukseen on tarjolla useita erilaisia menetelmiä, tekniikoita ja työvälineitä. Tässä luvussa on esitelty kaikki tässä työssä käytetyt työkalut.

4.1 Vesiputousmalli

Opinnäytetyön teossa käytettiin vesiputousmallin mukaista projektityömallia. Kuvassa 4.1 on esitetty vesiputousmallin vaiheet ja kulku.



Kuva 4.1 Vesiputousmalli (Oulun kauppaoppilaitos)

Vesiputousmallin keskeiset vaiheet ovat esitutkimus, määrittely, suunnittelu, toteutus, testaus ja käyttöönotto. Vesiputousmallissa vaiheet viedään läpi peräkkäin yhden kerran. (Huhtamäki 2005.)

Esitutkimuksen tehtävänä on tuottaa tietoa tietojärjestelmän kehittämisestä päättävälle sekä määrittää lähtökohdat mahdolliselle kehittämishankkeelle. Esitutkimuksen perusteella tehdään päätös järjestelmän kehittämisestä tai kehittämättä jättämisestä. (Oulun kauppaoppilaitos.)

Määrittelyvaiheessa selvitetään, mitä rakennettavan järjestelmän tulee tehdä. Siinä analysoidaan vaatimusmäärittelyn vaatimukset ja johdetaan niistä toiminnallinen määrittely. Vaatimustenmäärittely on dokumentti, johon on koottu kehitettävälle järjestelmälle asetetut vaatimukset eri sidosryhmiltä. Siinä ei oteta kantaa tekniseen toteutustapaan. Vaatimusten määrittely sisältää toiminnalliset vaatimukset, jotka määrittelevät mitä järjestelmän odotetaan tekevän sekä ei-toiminnalliset vaatimukset jotka määrittelevät järjestelmän reunaehdot (vastausajat, käytettävyys, rajoitteet). Vaatimustenmäärittely toteutetaan ennen toiminnallista määrittelyä. Toiminnallinen määrittely sisältää muun muassa järjestelmän jokaisen toiminnon yksityiskohtaisen kuvauksen, järjestelmän käsittelemien tietojen ja tietokantojen kuvaukset sekä järjestelmän rajapintojen kuvaukset. (Oulun kauppaoppilaitos.)

Suunnitteluvaiheessa suunnitellaan muun muassa ohjelmiston arkkitehtuuri, pääkomponentit ja käyttöliittymä. Määrittelyvaiheessa kuvataan ”mitä” järjestelmä tekee, suunnitteluvaiheessa ”miten” se sen teknisessä mielessä tekee. Suunnitteluvaihe jaetaan arkkitehtuuri- ja yksikkösuunnitteluun. Arkkitehtuuri-suunnittelussa kokonaisuus jaetaan itsenäisiin moduuleihin ja yksikkösuunnittelussa keskitytään moduulien sisäisiin rakenteisiin. (Turun yliopisto 2003.)

Toteutusvaiheessa ohjelmisto toteutetaan jollakin ohjelmointikielellä. Toteutuksen loppuvaiheessa moduulit kasataan toimivaksi kokonaisuudeksi. Toteutuksen tulee vastata sovellukselle asetettuja vaatimuksia ja sen tulee olla toiminnallisten ja teknisten määrittelyjen mukainen. (Oulun kauppaoppilaitos.)

Testauksen päätarkoituksena on virheiden löytäminen sovelluksesta. Yksi testautustavoista on moduulitestaus eli yksikkötestaus. Siinä etsitään virheitä yksittäisistä moduuleista ja testauksen suorittaa ohjelmistokehittäjä. (Oulun kauppaoppilaitos.)

Käyttöönotto voidaan suorittaa kun järjestelmä on testattu ja tuotteistettu. Tuotteistamiseen kuuluu muun muassa asennuspaketin kasaaminen, asennusohjeet ja käyttöohjeet. Käyttöönoton valmisteluun kuuluu olemassa olevien

tietojen siirtäminen uuteen järjestelmään ja ylläpitäjien kouluttaminen. (Oulun kauppappilaitos.)

Ylläpitovaihe on ohjelmiston elinkaaren pisin yksittäinen vaihe. Ylläpito kattaa tuotantokäytössä olevan järjestelmän virhekorjaukset, jatkokehityksen sekä muutokset. Ylläpitoa vaikeuttaa usein puutteellinen dokumentaatio. Jos järjestelmästä ylläpidetään useita eri versioita, pyritään ylläpitoa helpottamaan erilaisilla versionhallinta työkaluilla. (Oulun kauppappilaitos.)

4.2 Dynaaminen web-dokumentti

Web-dokumentit voivat sisältää muuttuvaa tietoa. Käyttäjän kannalta dokumentteihin on voitava hakea muuttuvaa sisältöä esimerkiksi tietokannasta tai tekstitiedostoista. Tiedon haku tapahtuu sen perusteella, mitä käyttäjä juuri sillä hetkellä haluaa. Toisaalta tiedon luonne itsestäänkin voi olla alati muuttuvaa, esimerkiksi säätiedot. Erilaisten organisaatioiden tietojärjestelmissä olevat tiedot ovat myös pääsääntöisesti muuttuvia. Ne vaativat ajan tasalla pysyäkseen helppokäyttöistä päivitettävyyttä. (Rantala 2002, 6.)

Tällaisiin tarpeisiin on syntynyt tekniikoita, joilla web-dokumentit voidaan luoda osittain tai kokonaan dynaamisesti. Dokumentit luodaan automaattisesti kulloisenkin lähtötilanteen pohjalta. Kun ohjelmoimalla pääsääntöisesti dynaamisesti luotu web-sivusto on luotu jotain tiettyä sovellusaluetta varten, voidaan puhua web-sovelluksesta. Minimissään web-sovellus voi koostua yhdestä pienestä PHP-komentosarjasta tai toisaalta useiden PHP-komentosarjojen kokonaisuudesta. Koska web-sovellukset toimivat pääsääntöisesti käyttäjän vuorovaikutuksen välityksellä, puhutaan myös interaktiivisista web-sivustoista. (Rantala 2002, 6.)

Dynaamisen web-dokumentin vastakohta on staattinen web-dokumentti. Staattisella web-dokumentilla tarkoitetaan sellaista sivua, jonka sisältö on tehty muuttumattomaksi. Käyttäjällä ei ole mahdollisuutta vaikuttaa sivun sisältöön. Tämän lisäksi staattisten web-dokumenttien päivitys vaatii jonkun erillisen ohjelman ja usein myös hieman HTML-kielen tuntemusta.

4.3 PHP

PHP on kieli, joka on ajan saatossa kasvanut ulos nimestään. Rasmus Lerdorf loi ensimmäisen PHP-version vuonna 1994. Tällöin PHP vielä koostui joukosta makroja, jotka autoivat ohjelmoijaa ylläpitämään henkilökohtaisia kotisivujaan (personal home pages) ja siitä PHP:n nimi syntyikin. Tämän jälkeen PHP:n ominaisuudet ovat lisääntyneet ja siitä on tullut oikea ohjelmointikieli, jolla voidaan hallita laajoja tietokantapohjaisia online-ympäristöjä. (Zandstra 2001, 20.)

Sitä mukaan, kun PHP:n ominaisuudet lisääntyivät, kasvoi myös sen suosio. PHP tunnetaan nykyään virallisesti nimellä PHP: HyperText Preprocessor (hypertekstin esikäsittelijä). Se on palvelinpuolen komentosarjakieli, jota kirjoitetaan tavallisesti HTML:n yhteydessä. Toisin kuin tavallinen HTML-sivu, PHP-komentosarjaa ei lähetetä suoraan asiakkaalle, vaan se jäsenellään PHP-ohjelman tai -moduulin toimesta. Komentosarjassa olevat HTML-elementit jätetään käsittelemättä, mutta PHP-koodi tulkitaan ja suoritetaan. Komentosarjassa oleva PHP-koodi osaa muun muassa tehdä kyselyjä tietokannasta, luoda kuvia, lukea tiedostoja ja kirjoittaa tiedostoihin sekä keskustella etäpalvelimien kanssa. PHP-koodin tulostus yhdistetään komentosarjassa olevaan HTML-koodiin ja tulos lähetetään käyttäjälle. (Zandstra 2001, 20.)

PHP:n on löysästi tyypitetty kieli mikä tarkoittaa, ettei sen muuttujien tietotyyppiä tarvitse etukäteen määritellä, toisin kuin joissain muissa ohjelmointikielissä. Muuttuja on varastopaikka, joka määritetään tallentamaan arvo. PHP-kielen muuttuja koostuu nimestä, jota edeltää taalamerkki (\$). Muuttujaan sijoitetaan arvo sijoitusoperaattorilla (=). Tietotyyppi vaikuttaa muistitilan tarpeeseen. Tietotyyppillä määritellään minkälaista tietoa muuttuja sisältää. Tieto voi olla muun muassa kokonaisluku, merkkijono tai totuusarvo. (Zandstra 2001, 46, 50-51.)

Esimerkki muuttujien alustuksesta ja käytöstä löytyy liitteestä 1.

4.4 MySQL ja SQL

MySQL on tietokannanhallintajärjestelmä, joka käsittelee relaatiotietokantoja. Tietokannanhallintajärjestelmä on järjestelmä, jolla voidaan hallita tietokantoja. Tietokannat ovat käytännössä katsoen tietovarastoja, joihin voidaan tallentaa tietoa ja joista voidaan hakea tietoa. Relaatiotietokannat ovat yksi tietokantatyypistä monien joukossa ja käytännössä eniten käytetyin tyyppi. Relaatiotietokannoissa tieto on järjestetty tauluihin, ja taulujen keskinäiset suhteet eli relaatiot ovat merkittävässä osassa. (Heinisuo & Rauta 2007, 37-38.)

MySQL on ruotsalaisen MySQL AB:n käsialaa, ja se oli alun perin tarkoitettu kyseisen konsultointiyrityksen sisäiseen käyttöön. MySQL on kehittynyt 90-luvun puolivälistä lähtien, ja tuotteen iän huomioon ottaen kyseessä ei ole mikään tähdenlento. MySQL on erinomaisen suorituskykyinen, ja sitä voidaan käyttää useista ohjelmointikielistä. MySQL on saatavilla ilmaiseksi GPL-lisenssillä. Mikäli MySQL-tekniikkaa käyttävää ohjelmistoa levitetään kaupallisesti, on MySQL-lisenssi ostettava erikseen. (Heinisuo & Rauta 2007, 38.)

MySQL-tietokannan kyselykielenä toimii SQL. SQL eli Structured Query Language on pitkälle standardoitu relaatiotietokantojen kysely- ja määrittelykieli. Sen rakenteet muistuttavat etäisesti englannin kieltä. Sen komentosanoja ovat muun muassa select, update, where ja order by. SQL on täysin merkkipohjainen, eli komennot voidaan kirjoittaa suoraan näppäimistöltä. Taulujen luonnin ja poiston lisäksi SQL tarjoaa komennot tauluihin tallennetun tiedon käsittelyyn. (Heinisuo & Rauta 2007, 97.)

SQL:n historia alkaa IBM:n laboratorion San Josesta Kaliforniasta, missä SQL kehitettiin 1970-luvun lopulla. Se kehitettiin alun perin IBM:n DB2:a (relaatiotietokannan hallintajärjestelmä) varten. SQL ei ole rakenteellinen kieli. Eirakenteinen tarkoittaa ”mitä” enemmän kuin ”miten”. Esimerkiksi SQL määrittelee, mitkä tiedot haetaan, poistetaan tai lisätään, eikä niinkään sitä, miten toimenpide tehdään. (Morgan, Perkins, Plew & Stephens. 2001, 3.)

MySQL käyttää asiakas-palvelin-arkkitehtuuria. Käytännössä tämä tarkoittaa, että MySQL toimii tietokoneella omana erillisenä sovelluksenaan sekä toimii palvelimena toisille ohjelmille. MySQL:n ja PHP:n tapauksessa tämä puolestaan tarkoittaa sitä, että tietokantaa ei käsitellä suoraan PHP-ohjelmasta käsin, vaan PHP:ssä otetaan yhteys MySQL-tietokantapalvelimeen. PHP lähettää palvelimelle SQL-kielisiä komentoja, ja MySQL voi lähettää komennosta riippuen vastineeksi PHP-sovellukselle takaisin tietokantaan tallennettua tietoa. (Heinisuo & Rauta 2007, 39-40.)

4.5 Hypertext Markup Language

HTML on WWW-dokumenttien eli WWW-sivujen rakenteen kuvaava sivunkuvauskieli. Se on kehitetty CERNissä 1990-luvun alkupuolella. CERN on Sveitsissä sijaitseva Euroopan ydintutkimuskeskus. HTML tulee sanoista HyperText Markup Language. (Sivut.web.)

HTML-dokumenttiin lisätään tekstisisällön lisäksi ohjeita tekstin osien eli elementtien loogisen rakenteen ja ulkoasun määrittelyä varten. Ohjeet kirjoitetaan dokumenttiin tarkasti määriteltyjen koodien eli tagien avulla. Tekstin lisäksi dokumentissa voi olla viittauksia kuviin, äänitiedostoihin ja muihin sovelluksiin sekä linkkejä. (Lehtonen 2001.)

HTML-dokumentit ovat aina ASCII-muotoisia tekstitiedostoja. Dokumenttien tiedostopääte on .html tai Dos-käyttöjärjestelmässä käytettävä .htm. HTML-dokumentteja voidaan kirjoittaa millä hyvänsä editorilla, joka tallentaa tiedostot ASCII-muotoon. (Lehtonen 2001.)

4.6 Cascading Style Sheets

CSS eli Cascading Style Sheets on yksinkertainen tyylikieli, jolla määritellään HTML, XHTML ja XML -dokumenttien ulkoasu ja esitystapa. Muiden tyylikielten tapaan CSS:n olemassaolon syynä on asiakirjan esittämiseen liittyvien ohjeiden erottaminen varsinaisesta asiakirjan sisällöstä ja rakenteesta. (KK Mediat.)

CSS:n kehitystyö alkoi 1990-luvun puolivälissä kun kävi ilmeiseksi, ettei HTML-kieli enää vastannut sille asetettuja vaatimuksia. Selainsodan (Netscape vs. Internet Explorer) riehussa huipussaan, molemmat selainvalmistajat lisäsivät HTML-kieleen selainkohtaisia piirteitä, joiden avulla voitiin määritellä muun muassa tekstin ulkoasu ja erilaisia korostuskeinoja. Lopputuloksena oli sivustojen toimimattomuus eri selaimilla sekä HTML-kielen jakaantuminen selainkohtaisiin versioihin. (KK Mediat.)

Ratkaisuksi tähän tilanteeseen kehitettiin CSS, jonka tavoitteena oli HTML-kielen rakenteellisuuden palauttaminen ja selainkohtaisten HTML-murteiden poistaminen. Tavoitteena oli myös erillisten muotoiluelementtien poistaminen HTML-kielestä sekä yhtenäinen, selaimesta, käyttöjärjestelmästä ja mediasta riippumaton tapa määritellä web-dokumenttien ulkoasu. Nämä tavoitteet ovat saavutettu hyvin pitkälti vasta XHTML-kielen myötä. (KK Mediat.)

4.7 JavaScript

Netscapen vuonna 1995 Navigator 2 -selaimen mukana julkaisema JavaScript 1.0 -kieli mahdollisti ensimmäistä kertaa interaktiivisten HTML-dokumenttien tuottamisen. Kielen alkuperäinen nimi oli LiveScript. Nimi kuitenkin vaihdettiin markkinoinnin vuoksi. JavaScript nimen valintaan vaikutti erityisesti Netscapen yhteistyö Java-kielen kehittäjän ja markkinoijan Sun Microsystemsin kanssa. (Peltomäki 2001, 2.)

JavaScript on tulkettava komentosarjakieli, joten se soveltuu hyvin lyhyiden, yksinkertaisten ja toistuvien ohjelmien tekoon. Tulkattavuus tarkoittaa sitä, että JavaScript-koodia ei käännetä ohjelman kirjoituksen jälkeen. JavaScript-ohjelmat, kuten muutkin komentosarjakieliä tulkataan samalla kun ne suoritetaan. Tämä mahdollistaa nopean ohjelmien teon ja testauksen sekä helpottaa ylläpitoa. Lisäksi se vähentää tietoverkon ruuhkaa, koska käännetyt ohjelmat ovat kooltaan huomattavasti suurempia kuin pelkkä JavaScript-koodi. (Kontio, Niskanen & Virimaa. 2001, 180.)

Oikeastaan on olemassa kaksi JavaScriptiä: tässä työssä käytetty asiakaspuolen JavaScript ja palvelinpuolen JavaScript. Asiakaspuolen JavaScriptiä voidaan kirjoittaa suoraan HTML-dokumenttiin script-elementin sisään ja se suoritetaan asiakaskoneessa. Palvelinpuolen JavaScript taas puolestaan sijaitsee palvelimella. Palvelinpuolen JavaScript käännetään suoritettavaksi tiedostoksi ja suoritetaan palvelimella. (Kontio ym. 2001, 180.)

Elementillä on joukko attribuutteja, joiden avulla voidaan reagoida käyttäjän toimiin. Näihin tapahtumiin voi JavaScriptillä ohjelmoida toimintaa eli tapahtumankäsittelijöitä. Hyvä esimerkki tapahtumasta on painikkeen painaminen. Tällöin tapahtuu onclick-tapahtuma. Tapahtumat ja tapahtumankäsittely liittyvät olennaisesti olio-ohjelmointiin ja JavaScript onkin olio-ohjelmointikieli. HTML-dokumenttia ja selaimen eri osia voidaan muokata eri olioiden avulla. (Kontio ym. 2001, 180-181.)

4.8 WampServer 2

WampServer on Windows-käyttöjärjestelmässä toimiva sovelluskehitysympäristö. WampServer on tarkoitettu web-sovellusten kehittämiseen ja suorittamiseen. Wamp tulee sanoista Windows, Apache, MySQL ja PHP. Nimi kertookin paljon WampServerin sisällöstä, sillä WampServer2 sisältää Apache-palvelimen, tuen PHP:lle ja MySQL:lle sekä phpMyAdminin, joka on MySQL-tietokannan hallintatyökalu.

4.9 SciTE-tekstieditori

PHP-ohjelmat ovat yleisesti tavallisia HTML-dokumentteja, joihin on upotettu PHP-ohjelmakoodia. Täten PHP-ohjelmia voidaan kirjoittaa millä tahansa tavallisella ASCII-tekstieditorilla. (Rantala 2002, 14.)

SciTE on tekstieditori, joka sisältää monista muista editoreista tuttuja toimintoja, kuten välilehdet useille tiedostoille, värikoodauksen ja joidenkin ohjelmointikielten funktioiden automaattitäydennyksen. Tuettuja ohjelmointikieliä SciTEssä on noin 30. Ohjelmakoodin kääntäminen ja testaus on tehty myös helpoksi sillä

valikoista löytyvillä build- ja run-komennoilla lähdekoodia voidaan testata suoraan editorista. SciTE on saatavana Linuxille ja Windowsille. (Tietokone 2008.)

Liitteen 1 koodiesimerkki on kuvankaappaus SciTE-ohjelmasta.

5 PHP TOTEUTUSVÄLINEENÄ

Erilaisia tekniikoita dynaamisten web-sovellusten tekemiseen on useita. PHP:n etuna on se, että se on suunniteltu nimenomaan web-sovellusten tekemiseen. PHP:ssä käytetyt ratkaisut mahdollistavat nopean ohjelmistokehityksen. Selaimelta saatava tieto on helposti suoraan käytettävissä. Heikosti tyyhitettynä kielenä sen muuttujien tyyppiä ei tarvitse määritellä. Yksinkertaisuudestaan huolimatta PHP on tehokas ja monipuolinen, ja avoimena ohjelmistona se on saatavilla vapaasti ilmaiseksi. (Rantala 2002, 13.)

5.1 PHP:n valinnan perusteet

Kun Hyvän Olon Päiväkirjan toteutusta alettiin suunnitella kesällä 2009, oli toteutustekniikaksi tarjolla kaksi vaihtoehtoa, ASP.net ja PHP. Tekniikaksi valittiin PHP-kieli, koska siitä minulla on enemmän kokemusta. PHP on kieli, jolla opin ohjelmoimaan. Samalla halusin oppia käyttämään PHP:tä paremmin web-sovellusten teossa.

Olin tehnyt muutamia töitä ASP.net-tekniikalla juuri ennen Hyvän Olon Päiväkirjaa. Pidän ASP.net sovelluskehityksestä todella paljon. Halusin kokeilla, pystyykö tai edes kannattaako PHP-kielillä toteuttaa dynaamisia web-sovelluksia samaan tapaan kuin ASP.net-tekniikalla. Tällä tarkoitan tässä tapauksessa sitä, että HTML-kieli ja sivun tulostus pidetään erillään muusta koodista.

Useasti näkee sivuja, jotka ovat tehty niin, että HTML-muotoilut ja PHP-koodi ovat keskenään sekaisin, ja koodi sekä tulostukset ovat kirjoitettu ylhäältä alaspäin siinä järjestyksessä kuin niitä sivun tulostuksessa tarvitaan. Tämä johtaa usein siihen, että kun koodin välistä poistaa yhdenkin rivin, koko sivun tulostus

hajoaa. Halusin tehdä sivuista mahdollisimman modulaariset, jolloin yhden osan poisto ei välttämättä riko koko sivua ja niin, että osat ovat mahdollisimman uudelleen käytettäviä.

Nykyään on ollut suuntauksena, ettei eri kieliä sekoiteta toisiinsa. Kun koodi eli sivun toiminnallisuus ja tulostukset eli esimerkiksi HTML-muotoilut pidetään erossa toisistaan, tulee sivujen toteuttamisesta, tulkinnasta ja muunneltavuudesta helpompaa. Tämän lisäksi tietokantakyselyt olisi vielä hyvä erottaa toiminnallisuudesta. Tämä edistäisi omalta osaltaan sivujen ylläpidettävyyttä. Kun esimerkiksi halutaan vaihtaa sovelluksen tietokantajärjestelmää, onnistuu se helpommin, jos tietokantakyselyt löytyvät yhdestä paikkaa, sen sijaan, että ne olisi sijoitettu useaan tiedostoon koodin sekaan.

Koska PHP kykenee kaikkeen edellä mainittuun ja paljon muuhunkin (kuten olio-ohjelmointiin), olisi näitä suuntauksia syytä seurata jo pelkästään sovellusten toteutustapojen yhdenmukaisuuden kannalta. Pitää kuitenkin pitää mielessä, että aina ei voida edellä mainittuja ratkaisuja suoraan noudattaa, esimerkiksi toteutusteknisten rajoitteiden vuoksi. Toteutustapoja voidaan kuitenkin soveltaa kulloiseenkin projektiin parhaan mukaan.

5.2 Aikaisemmin kohtaamani ongelmat

PHP-kielessä ei ole olemassa yhtä standardia, jonka mukaan laajempia PHP-sovelluksia kannattaisi tehdä. Tämä on johtanut siihen, että standardeja on yhtä monia kuin tekijöitäkin. PHP-sovellukset ovat koodiltaan ja toteutukseltaan usein hyvinkin erinäköisiä, verrattuna esimerkiksi Microsoftin työkaluilla ja tekniikoilla tehtyihin sovelluksiin, jotka tekijästä riippumatta noudattavat keskenään melko samanlaista kaavaa. Ehkä olio-ohjelmoinnin mukaan tulo myös PHP-sovelluksiin yhtenäistää PHP-sovellusten toteutustapoja.

5.3 PHP:n ja asiakaspuolen tekniikoiden yhdistäminen

Asiakaspuolen tekniikoilla tarkoitetaan sellaisia tekniikoita jotka ladataan WWW-sivun suorituksen yhteydessä ja suoritetaan käyttäjän omalla koneella. Tällaisia tekniikoita ovat esimerkiksi Ajax, Flash tai JavaScript.

Jos PHP-sovellusten käytettävyys halutaan hioa huippuunsa, kannattaa harkita asiakaspuolen tekniikoiden käyttöä sivujen toteutuksessa. Asiakaspuolen tekniikoiden yhdistäminen PHP-sovelluksiin on yleensä melko helppoa. Niitä käytettäessä on kuitenkin muistettava, että käyttäjä voi halutessaan helposti estää kyseisten tekniikoiden toimimisen selaimessaan. Myöskään kaikki selaimet eivät välttämättä tue kaikkia asiakaspuolen tekniikoita. Sovellukset pitääkin suunnitella niin, että niiden koko toiminnallisuus perustuu asiakaspuolen tekniikoiden käyttöön (esimerkiksi YouTube videopalvelu) tai niin, että sovellusten käytettävyys ei kärsi mikäli tekniikat eivät ole käytössä. Hyvän Olon Päiväkirjan pääsivuilla asiakaspuolen tekniikoita päätettiin olla käyttämättä, mutta pääsivujen jatkokehityksessä ja ylläpitosivujen toiminnallisuudessa niitä voidaan kuitenkin käyttää, mikäli on tarvetta.

6 SUUNNITTELUN LÄHTÖKOHDAT

Suunnittelussa lähdettiin liikkeelle ensisijaisesti asiakkaan toiveista. Asiakkaalla ei oikeastaan ollut mitään toiveita sisällön suhteen, vaan sain melko vapaat kädet ylläpitosivujen sisällön suunnitteluun. Asiakkaan toiveet koskivat lähinnä ylläpitosivujen käytettävyyttä. Käyttöliittymän tuli olla selkeä ja helppokäyttöinen. Tiedon syötön ja muokkaamisen piti onnistua helposti ja keneltä tahansa. Lisäksi asiakas toivoi, ettei ylläpitäjä pääse millään tavalla selailemaan käyttäjien syöttämiä tietoja.

Suunnittelun toisena lähtökohtana toimi sivuston toteutuksessa käytetyt tekniikat sekä tekniikoiden asettamat rajat. PHP on kielenä ja toteutustekniikkana melko pelkistetty, moniin muihin tekniikoihin verrattuna. Myös toteutusvälineenä

käytettävä tekstieditori ohjasi suunnittelua siihen suuntaan, että itse toteutus olisi mahdollisimman yksinkertainen.

Koska toteutin myös Hyvän Olon Päiväkirjan pääsivut, oli minulla melko selkeä kuva siitä, mitä ylläpitosivujen pitäisi sisältää. Suurin osa ideoista syntyi jo pääsivujen suunnittelun ja toteutuksen aikana. Nuo ideat toimivat suunnittelun pohjana.

7 HYVÄN OLON PÄIVÄKIRJAN YLLÄPITOSIVUJEN TOTEUTUKSEN VAIHEET

Opinnäytetyön teossa käytetyn vesiputousmallin tavanomaista kulkua muutettiin niin, että määrittely ja suunnittelu yhdistettiin. Tuota vaihetta kutsutaan tässä luvussa suunnitteluvaiheeksi. Tämän lisäksi työt jaettiin suunnitteluvaiheessa moduuleihin, jotka toteutusvaiheessa jaettiin vielä pienempiin moduuleihin. Moduulit testattiin heti niiden valmistuttua. Nämä ratkaisut vauhdittivat työn etenemistä.

7.1 Esitutkimus

Esitutkimus tehtiin joulukuussa 2009. Esitutkimuksessa kuvailtiin tuoteidea, työle asetettavat vaatimukset, projektin organisaatio ja aikataulu sekä projektiin liittyvät mahdolliset ongelmat.

Tuoteideassa esiteltiin Hyvän Olon Päiväkirja. Vaatimuksissa käytiin läpi ylläpitosivuille halutut ominaisuudet. Ylläpitosivujen lisäksi esitutkimusvaiheessa mukana olivat myös ohjelman ensimmäisen version toteutuksen ja käytön aikana esille tulleet jatkokehitysideat. Projektioorganisaatioon nimettiin tekijän ja yrityksen edustajan lisäksi tuntiopettaja Jouni Könönen, joka toimi opinnäytetyön ohjaajana koulun puolelta. Aikatauluksi asetettiin Saimaan ammattikorkeakoulun opinnäytetyöaikataulu, joka tähtäsi valmistumiseen kesäkuussa 2010. Esitutkimuksessa ilmenneissä ongelmassa mainittiin Hyvän Olon Päiväkirjan käyttöönoton hieman viivästyneen muutosten ja uusien ominaisuuksien takia. Viivästy-

miset haluttiin tällä kertaa välttää huolellisesti suoritettavalla suunnittelulla. Esi- tutkimuksen jälkeen projekti todettiin toteuttamiskelpoiseksi ja siirryttiin seuraavaan vaiheeseen.

7.2 Suunnittelu

Ennen suunnitteluvaiheeseen siirtymistä pidettiin aloituskokous, jossa oli paikalla koko projektiorganisaatio. Aloituskokous pidettiin joulukuun 9. päivä 2009. Aloituskokouksessa päätettiin projektin kulusta ja tarkemmista aikatauluista.

Itse suunnitteluvaihe aloitettiin kartoittamalla kaikki ne tehtävät, jotka projektiin kuuluivat. Tehtävät olivat ohjelman ylläpitosivujen tai itse pääsivujen ominaisuuksia ja toiminnallisuuksia. Tässä vaiheessa tehtäviä kertyi jo niin paljon, että työtä päätettiin rajata. Työtä rajattiin niin, että projektissa keskityttiin vain ylläpitosivuihin. Jatkokehitysideat jätettiin pois. Ratkaisuun päädyttiin, jotta projekti pysyisi aikataulussaan.

Kartoitetut tehtävät olivat moduuleita, jotka pystyttäisiin toteuttamaan ja testaamaan itsenäisesti. Kun nämä moduulit oli tarkasti kartoitettu, kuvattiin kukin niistä erittäin lyhyesti ja karkeasti. Yksi moduuli oli esimerkiksi ylläpitosivujen Käyttäjien hallinta -sivut. Kuvauksessa määriteltiin, että kyseisten sivujen pitää sisältää muun muassa ominaisuudet Lisää käyttäjä ja Muokkaa käyttäjiä. Tämän jälkeen kuvailtiin, kuinka kyseisten ominaisuuksien pitää toimia. Tätä samaa kaavaa sovellettiin myös muihin moduuleihin.

Vaikka ylläpitosivut käyttivät tietokantaa, ei tietokantaa tarvinnut suunnitella tässä vaiheessa. Hyvän Olon Päiväkirjan tietokanta oli suunniteltu ja toteutettu jo pääsivuja toteutettaessa. Tietokanta suunniteltiin jo tuolloin ylläpitosivuja silmällä pitäen, joten suurempiin muutoksiin ei ollut tarvetta.

Ylläpitosivujen graafinen ilme noudatti samaa kaavaa kuin pääsivutkin, joten sitä ei tarvinnut suunnitella erikseen.

Suunnitelmat hyväksyttiin asiakkaalla ja sovittiin, että jatkokehitysideoita toteutetaan, jos niille jää aikaa.

7.3 Toteutus

Toteutusvaiheen aluksi kartoitettiin erilaisia tapoja toteuttaa PHP-sovelluksia perehtymällä kirjallisuuteen ja selaillemalla Internetiä. Tämän lisäksi testattiin erilaisia työkaluja joilla sivut voitaisiin toteuttaa. Testauksessa olivat mukana ohjelmointiympäristöt NetBeans ja Eclipse sekä HTML-editorit Macromedia HomeSite ja Evrsoft 1stPage 2006. Lisäksi punnittiin millä tavalla ylläpitosivut toteutettaisiin. Vaihtoehtoina olivat esimerkiksi oliopohjainen ratkaisu tai ASP.net-tyylillä toteutettu ratkaisu, jossa koodit ja sivun tulostus ovat eri tiedostoissa.

Hyvän Olon Päiväkirjan pääsivut suunniteltiin ja toteutettiin kiireessä osittain muiden töiden ohella. Tästä muistuttaa myös sivujen lähdekoodi, joka on tyyliltään lähempänä kiireessä tehtyä demoversiota kuin harkiten rakennettua modulaarista kokonaisuutta. Hyvän Olon Päiväkirjan pääsivut toteutettiin ilman olioita perinteisellä tekstieditorilla. Ylläpitosivujen toteutustavan ei haluttu liikaa poikkeavan pääsivujen toteutustavasta, joten niissä päädyttiin kompromissiin. Ylläpitosivut toteutettiin tekstieditorilla, mutta sivurakenne tehtiin niin, että toiminnot ja sivun tulostus pyrittiin pitämään erillään, mutta kuitenkin niin, että ne sijaitsevat samassa tiedostossa. Tästä lisää luvussa 8.1, Sivurakenne.

Kun toteutustapa oli päätetty, pystytettiin työasemat, joilla työ tehtiin. Työasemina toimi kaksi tietokonetta. Toinen oli pöytätietokone, jossa oli käyttöjärjestelmänä Windows XP, ja toinen oli kannettava tietokone, jossa oli käyttöjärjestelmänä Windows 7. PHP-sovelluksen kehitystä varten tarvittiin Windows käyttöjärjestelmässä toimiva palvelin, jolta sivuja ajettiin, tietokanta, johon tiedot tallennettiin, sekä tuki PHP:tä ja MySQL:ää varten. Tähän tarkoitukseen oli olemassa WampServer 2. Pöytätietokoneessa WampServer oli jo valmiina, mutta kannettavaan tietokoneeseen se oli vielä asennettava. WampServerin asennus Windows 7:n käyttöjärjestelmään onnistui vaivattomasti.

Toteutuksessa käytettiin PHP:n versiota 5.3.0 ja MySQL:n versiota 5.1.36.

Ylläpitosivut koostuivat kahdeksan osa-alueen hallintasivuista. Osa-alueita olivat

- käyttäjät
- sisältö
- ravinto
- liikunta
- uni
- stressi
- tupakka
- tietoisuus.

Kaikki osa-alueet, sisältöä lukuun ottamatta, sisälsivät lisäys-, muokkaus- ja poisto-toiminnot sekä tietojen selailu ja etsintä mahdollisuudet. Jokaisella osa-alueella oli kuitenkin omat erityispiirteensä, joten yhden osa-alueen toiminnallisuutta ei voitu suoraan käyttää toisen osa-alueen hallintasivuilla. Lisäksi joillakin osa-alueilla oli vielä muutamia muitakin toimintoja.

Ylläpitosivujen toteutus aloitettiin Käyttäjien hallinta -sivuilla. Sivut sisälsivät perustoimintojen lisäksi mahdollisuuden lähettää muistutuksia ja yksityisviestejä käyttäjille.

Tässä vaiheessa suunnitteluvaiheessa jaetut moduulit jaettiin vielä pienempiin moduuleihin. Esimerkiksi Lisää käyttäjä -toimintoon kuului Lisää- ja Poista-painikkeet. Kummankin painikkeen toiminnallisuus toteutettiin ja testattiin itsenäisesti, vaikka ne kuuluivat samaan kokonaisuuteen.

Käyttäjien lisäys toteutettiin niin, että tietokantaan pystyi lisäämään useamman käyttäjän kerrallaan. Lisättävien käyttäjien sähköpostiosoitteet syötettiin pilkulla eroteltuina tekstikenttään. Kun tiedot oli syötetty, ohjelma tarkisti, että osoitteet olivat oikeassa muodossa ja että ne eivät olleet jo käytössä sovelluksessa. Jos osoitteet olivat kunnossa, niille jokaiselle luotiin oma satunnainen salasana.

Tämän jälkeen osoite, joka myös toimi käyttäjätunnuksena, ja salasana tallennettiin tietokantaan, ja lähetettiin tervetulo-viestin kera tallennettuun sähköpostiosoitteeseen.

Jos tietokantaan tallennettiin useampia käyttäjiä kerallaan, suoritettiin tietokantaan tallennus transaktiona. Transaktio toimii niin, että tietokantaan tallennetaan yksi tieto kerrallaan. Jos tiedon tallennus onnistuu, tallennetaan seuraava ja niin edelleen. Jos yhdenkin tiedon tallennus epäonnistuu, lopetetaan tietojen tallennus ja perutaan kaikki transaktiossa jo mahdollisesti tallennetut tiedot. Ratkaisuun päädyttiin, jotta olisi helpompi seurata mitä tietoja tietokantaan tallennettiin. Jos transaktio epäonnistui, pystyttiin ylläpitäjälle ilmoittamaan, että mitään tietoja ei tallennettu sen sijaan, että joidenkin tietojen tallennus onnistui ja toisten ei. Tämä ratkaisu yksinkertaisti toteutusta ja sitä käytettiin myös muissa ylläpitosivujen toiminnoissa.

Kun Käyttäjien hallinta -sivut oli tehty, pystyi niiden toiminnallisuuksia hyödyntämään kaikilla muillakin sivuilla. Esimerkiksi eri osa-alueiden tietojen selailu- ja muokkaussivut olivat joitakin poikkeuksia lukuun ottamatta lähes identtiset. Sivut saatiin muokattua halutun osa-alueen käyttöön vaihtamalla tietokantakyselyyn taulun nimi ja haettavat tiedot sekä määrittelemällä tulostettavan taulukon koko ja sisältö. Poikkeusten takia sovellusta ei kuitenkaan voitu laittaa käyttämään yhtä ja samaa selailu- ja muokkaussivua kaikille osa-alueille, vaan kaikille piti tehdä omat yksilölliset sivunsa. Kuvassa 7.1 on esimerkki Liikunta-osa-alueen selailu- ja muokkaussivusta, joka sisältää muun muassa muokkaa- ja poista-toiminnot.

Hallinta: Muokkaa liikuntalajeja (Perjantai 07.05.2010) [\(Ohje\)](#)

Lajeja yhteensä: 73

Etsi: Sivulla näytetään: Lisääjä: Käyttö:

	Liikunta	Kulutus kcal/kg/min	Kestävyys (reipas)	Kestävyys (rasittava)	Lihaskunto	Lisääjä		
<input type="checkbox"/>	Kävely, rauhallinen 4km/h	0.05	kyllä	ei	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Kävely, reipas 6km/h	0.067	kyllä	ei	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Leikkiminen lasten kanssa, kohtuullinen rasitus, kävelyä	0.067	kyllä	ei	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Luistelu	0.123	ei	kyllä	kyllä	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Lumen luonti lapiolla tai lumikolalla	0.1	kyllä	ei	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Lumikenkäkävely	0.134	ei	kyllä	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Maalaaminen	0.075	kyllä	ei	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Melonta, kohtuullinen rasitus	0.117	ei	kyllä	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Melonta, rasittava	0.201	ei	kyllä	ei	admin	muokkaa	<input type="button" value="Poista"/>
<input type="checkbox"/>	Melonta, rauhallinen	0.067	kyllä	ei	ei	admin	muokkaa	<input type="button" value="Poista"/>

<< Edellinen 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Seuraava >>

Kuva 7.1 Muokkaa liikuntalajeja -sivu

Työt etenivät yksi osa-alue kerallaan. Kun tietyn osa-alueen kaikki toiminnot oli toteutettu ja testattu, siirryttiin seuraavaan osa-alueeseen. Suurempia ongelmia toteutusvaiheessa ei ilmennyt, vaan työt etenivät hyvällä tahdilla eteenpäin.

Ensimmäinen todellista päänvaivaa aiheuttanut ongelma kohdattiin tietoiskut-osa-alueella toteutettaessa. Tietoiskut ovat artikkeleita, joissa käsitellään terveyteen ja hyvinvointiin liittyviä asioita. Ylläpitäjälle haluttiin antaa mahdollisuus muokata lisäämiään artikkeleitaan niin, että tekstiä pystyi muotoilemaan, ja tekstin sekaan pystyi lisäämään linkkejä ja kuvia. HTML-kielen peruselementeillä ja PHP:llä tätä ei kuitenkaan ollut mahdollista toteuttaa tehokkaasti. Tiedon syötössä yleensä käytetty HTML-kielen textarea-elementti ei näytä muotoiltuja tekstejä eikä kuvia, vaan pelkän tekstin ja tekstin muotoiluun käytetyt tagit. HTML-kieltä tuntemattomia tekstin seassa olevat tagit usein hämmentävät, joten niiden näyttämistä suoraan käyttäjälle kannattaa välttää.

Tekstien muotoiluun web-sovelluksissa löytyy useita työkaluja. Yksi niistä on Cross-Browser Rich-Text Editor, tai lyhyemmin RTE. RTE on niin kutsutun ri-

kastetun tekstin muotoiluun tarkoitettu editori. RTE on saatavilla myös ilmaiseksi ja sitä saa käyttää vapaasti tietyin ehdoin. RTE:llä on lähes kaikki samat ominaisuudet kuin perustekstinkäsittelyohjelmillakin. RTE on toteutettu JavaScript-komentosarjakiielellä, ja sitä on erittäin helppo käyttää myös PHP-sovelluksissa.

Koska RTE säilyttää tekstin muotoilut, tulee liitetyn tekstin mukana paljon muutakin kuin pelkkä teksti. Hyvänä esimerkkinä toimii tapaus, jossa käyttäjä kopioi Microsoft Word -dokumentista yhden sanan ja liittää sen Mozilla Firefox-selaimessa näytettävän web-sovelluksen RTE-tekstikenttään. Todellisuudessa käyttäjä kopioi yli tuhat sanaa eli noin 18 000 merkkiä, jotka Word-dokumenttiin liitettynä tarkoittaa reilua kuutta sivua tekstiä. Noin suuren merkkimäärän tallentaminen tietokantaan ei ole järkevää. Merkkien määrä johtuu siitä, että Word-dokumentti sisältää paljon erilaisia muotoiluja, joita ei ulospäin näytetä. Noita muotoiluja ei ainakaan RTE:n tässä työssä käytetty versio osannut poistaa. Ongelmaa pyrittiin lieventämään siten, että käyttäjä ohjattiin kopioimaan tekstinsä HTML-kielen textarea-elementtiin, joka poistaa muotoilut tekstistä liitettäessä. Kun käyttäjä halusi muotoilla tekstiä, pääsi hän painiketta painamalla käyttämään Rich Text Editoria. Tulevaisuudessa editoriin voitaisiin lisätä ominaisuus, jolla estettäisiin käyttäjää liittämästä tekstiä suoraan editoriin.

Vaikka ylläpitosivujen graafinen ilme ja muotoilu noudattivatkin samaa kaavaa kuin pääsivut, luotiin ylläpitosivuja varten oma CSS-tyylitiedostonsa. Tyylitiedosto sisälsi muutamia luokkia ja muotoiluja, joita ei pääsivujen tyylitiedostosta löytynyt. Uutta grafiikkaa ei tarvinnut tehdä lainkaan, sillä kaikki tarvittavat graafiset-elementit löytyivät jo sovelluksesta.

Työpäivät venyivät pisimmillään jopa 13 tuntiseksi. Noin pitkiä työpäiviä tehtäessä taukojen merkitys korostui. Kun jotain ongelmaa oli pohdittu useita tunteja tuloksetta, muutaman kymmenen minuutin ulkoilun jälkeen ongelmaan löytyi usein kaivattu ratkaisu. Välillä oli hyvä siirtyä tietokoneen ääreltä kokonaan pois, jonnekin jossa pystyi rauhassa pohtimaan ratkaisuja tai vain siirtämään ajatukset muualle ja huilaamaan hetken. Mitä vireämpi mieli oli, sitä paremmin työt edistyivät.

7.4 Testaus

Aina kun yhden osa-alueen yhden toiminnallisuuden yksi moduuli saatiin valmiiksi, se testattiin välittömästi. Testaus suoritettiin sekä lähdekooditasolla että käyttöliittymän kautta.

Useimmat lähdekooditasolla testattavat moduulit testattiin sillä periaatteella, että aluksi testattiin muu toiminnallisuus, ja kun oltiin varmoja, että moduulin käsittelemä tieto oli oikeanlaista, testattiin tietokantaan kohdistuva toiminnallisuus. Tämä koski tietenkin vain niitä moduuleita, jotka olivat tietokannan kanssa tekemisissä.

Yksi moduuli oli esimerkiksi Käyttäjät-osa-alueen Muokkaa käyttäjiä -sivun Kyllä-painikkeen toiminnallisuus. Kyllä-painikkeella varmistettiin, että valitut käyttäjät haluttiin poistaa sovelluksen tietokannasta. Moduuli testattiin niin, että aluksi varmistettiin, että painikkeen painallus valitsi oikeat käyttäjät poistettavaksi. Tämän jälkeen testattiin, että tietokantakysely, jolla käyttäjät poistetaan, oli oikeanlainen. Kun edellä mainitut vaiheet olivat kunnossa, voitiin suorittaa käyttäjän poisto ja testata koko moduulin toiminnallisuus. Lopuksi tarkistettiin, että valitut käyttäjät oli varmasti poistettu tietokannasta.

Muita testauskohteita olivat muun muassa tekstikentät, joihin sai syöttää vain tietyn tyyppistä ja tietokantaa vahingoittamatonta tietoa. Tietokantaa vahingoittamattomalla tiedolla tarkoitetaan sellaista tietoa, joka ei sisällä tiettyjä merkkejä, joita tietokantakysely ei saa sisältää mahdollisten tietoturvariskien estämiseksi. Tällaisia merkkejä ovat esimerkiksi lainaus- ja heittomerkit.

Joihinkin tekstikenttiin sai syöttää esimerkiksi vain pelkkiä kokonaislukuja. Ennen kuin nämä tiedot tallennettiin, sovelluksen piti tarkistaa, että tiedot varmasti olivat kokonaislukuja. Jos syötetyt tiedot olivat muita kuin kokonaislukuja, ei tietoja tallennettu ja käyttäjälle ilmoitettiin virheestä. Kyseistä toiminnallisuutta testattiin syöttämällä tekstikenttiin erilaisia merkkijonoja ja tarkkailemalla, tunnistaaako sovellus ne. Syötettyjä merkkijonoja saattoivat olla esimerkiksi 13, 6.1, 2k1, yksi, "12", #a%&78yuu ja niin edelleen. Edellä luetelluista merkkijonoista

vain ensimmäinen, eli 13, oli oikeanlainen kokonaisluku. Jos sovellus tallensi sen ja ilmoitti virheestä muiden merkkijonojen kohdalle, se toimi oikein.

Yksittäisten moduulien testauksen lisäksi testattiin kokonaisuuksia. Yksi kokonaisuus oli esimerkiksi Muokkaa käyttäjiä -sivu, jonka moduulien piti toimia keskenään oikein ja muodostaa haluttu kokonaisuus. Sovelluksen piti osata siirtyä toiminnallisuudesta toiseen aiheuttamatta ristiriitoja suorittamatta turhaa koodia. Tätä testattiin suorittamalla kaikkia mahdollisia toiminnallisuuksia, joita sivu sisälsi, ja seuraamalla, että sovellus käyttäytyi suunnitellulla tavalla.

Ennen sovelluksen käyttöönottoa, jota ei vielä tätä raporttia kirjoitettaessa ole suoritettu, suoritetaan järjestelmätestaus. Järjestelmätestauksessa testataan käyttöliittymän kautta koko sovellus, tietokanta ja se laitteisto, jolla sivustoa ylläpidetään.

7.5 Käyttöönotto

Käyttöönottoa silmällä pitäen kaikista tietokantaan tehdyistä muutoksista pidettiin kirjaa. Suurena haasteena käyttöönotossa tulee olemaan se, kuinka kyseiset muutokset saadaan tehtyä sovelluksen nykyiseen tietokantaan vahingoittamatta tietokannan rakennetta ja tietoja.

8 YLLÄPITOSIVUJEN ESITTELY

Kaikki ylläpitosivuihin kuuluvat tiedostot sijaitsevat Hallinta-kansiossa, joka sijaitsee sivujen juuressa. Tuo kansio sisältää myös muita, pääsivuihin kuuluvia tiedostoja kuten funktiot ja muuttujat. Kaikki tiedostot, muutamaa poikkeusta lukuun ottamatta, ovat PHP-tiedostoja.

Ylläpitosivujen osoiterivillä kutsuttavat ja käyttäjälle näkyvät tiedostot ovat nimetty tyyliin h_osa-alue_toiminto. Nimen alussa oleva h-kirjain viittaa hallintaan, osa-alueessa viittaa, mihin osa-alueeseen tiedosto kuuluu, ja toiminto viittaa siihen, mitä tiedostossa tehdään. Esimerkiksi käyttäjän lisäämiseen tarkoitettu

tiedosto on nimeltään h_kayttaja_lisaa.php. Tämä nimeämistapa helpottaa tiedostojen tunnistamisessa ja erottamisessa muista tiedostoista.

Edellä mainittujen tiedostojen lisäksi Hallinta-kansio sisältää muita, kansioita joihin on sijoitettu ylläpitosivuihin kuuluvia käyttäjälle suoraan näkymättömiä tiedostoja. Kansioita ovat function, include, rte, stylesheet ja tmp. Function-kansio sisältää ylläpitosivujen funktiot ja muuttujat. Include-kansiossa ovat tiedostot joista tulostettava sivu rakennetaan. Rte-kansiossa on Rich Text Editoriin kuuluvat tiedostot. Stylesheet-kansiossa on ylläpitosivujen oma tyylitiedosto. Tmp-kansioon tallennetaan muun muassa esikatseluissa käytetyt väliaikaiset tiedostot, jotka tuhoetaan, kun tiedot lopullisesti tallennetaan. Tiedostojen jakaminen eri kansioihin selkeyttää sivuston rakennetta ja helpottaa tiedostojen löytämistä.

8.1 Sivurakenne

Osoiterivillä kutsuttava käyttäjälle tulostettava sivu koostuu useasta tiedostosta. Tiedostot ovat sisällytetty tulostettavaan sivuun PHP-kielen `require_once()`-funktioilla. `Require_once()`-funktio liittää sille parametriksi annetun tiedoston tulostettavaan sivuun. Tiedostot sisällytetään kirjoittamalla liitettävän tiedoston nimi ja sijainti parametriksi sulkujen sisälle heti funktion nimen perään, esimerkiksi `require_once(kansio/tiedosto.php)`. Näin liitetyn tiedoston sisältö tulee osaksi kutsuttavan sivun sisältöä.

Liitteessä 1 on pelkistetty esimerkki kutsuttavasta sivusta. Kutsuttava sivu rakentuu siten, että kutsuttavan sivun alkuun on sisällytetty pääsivujen `ot.php`- ja `muuttujat.php`-tiedostot. Näin kutsuttava sivu saa käyttöönsä kyseisten tiedostojen sisällön. Tämän jälkeen alustetaan sivun omat muuttujat. Muuttujien alustuksien jälkeen suoritetaan sivujen toiminnallisuus. Toiminnallisuuksista ensimmäisinä suoritetaan sivulla olevien painikkeiden painalluksista suoritettavat toiminnot. Seuraavaksi suoritetaan painikkeiden painalluksista riippumattomat toiminnallisuudet. Niistä ensimmäisenä suoritetaan ne toiminnallisuudet, jotka suoritetaan vain silloin, kun sivulle saavutaan ensimmäisen kerran. Viimeisenä suoritetaan ne toiminnallisuudet, jotka suoritetaan jokainen kerta,

kun sivu ladataan. Sivun tulostuksessa tarvittavat tiedot ja elementit luodaan täällä. Toiminnallisuuden jälkeen alkaa sivun tulostus. Kuvassa 8.1 on kutsuttava sivu tulostettuna ja osiin jaettuna.

Saimaan ammattikorkeakoulu
Saimaa University of Applied Sciences

Etusivu Käyttäjät Sisältö Ravinto Liikunta Uni Stressi Tupakka Tietoiskut Ohje Kirjautuu ulos

> Pääsivu
> Lisää univinkki
> **Muokkaa univinkkejä**

Hallinta: Muokkaa univinkkejä (Sunnuntai 09.05.2010) (Ohje)

Tietoiskuja yhteensä: 36

Etsi: Etsi Sivulla näytetään: 10 Järjestys: Väri (nouseva) OK

Teksti	Väri		
<input type="checkbox"/> Alavartalon ja jalkojen lämpimänä pitäminen edistää nukahtamista, jalkakäyry tekee hyvää.	keltainen	muokkaa	Poista
<input type="checkbox"/> Jokaista valvottua 3-4 tuntia kohden tarvitaan keskimäärin 1h unta.	keltainen	muokkaa	Poista
<input type="checkbox"/> Jos sinulla on nukahtamisvaikeuksia, nouse ylös sängystä ja tee jotain rauhoittavaa, esim. lue kirjaa. Palaat sänkyyn vasta, kun sinua alkua nukkumaan.	keltainen	muokkaa	Poista
<input type="checkbox"/> Juo säästeliäästi ennen nukkumaan menoa, jotta yöuni ei ole katkonnaista wc-hädän vuoksi.	keltainen	muokkaa	Poista
<input type="checkbox"/> Kun nukkuu, niin ei väsytä.	keltainen	muokkaa	Poista
<input type="checkbox"/> Mene sänkyyn vasta kun sinua väsyttää ja olet uninen.	keltainen	muokkaa	Poista
<input type="checkbox"/> Nauti nautintoaineita ennen nukkumaan menoa harkiten. Muista, että kahvi, vahva tee, kaakao ja tumma suklaa piristävät.	keltainen	muokkaa	Poista
<input type="checkbox"/> Nuku pimeässä.	keltainen	muokkaa	Poista
<input type="checkbox"/> Ole uskollinen itsellesi, noudata oman unen tarvetta.	keltainen	muokkaa	Poista
<input type="checkbox"/> Pyritä sänky nukkumiselle. Älä tee työasioita tai katso telkkaria sängyssä.	keltainen	muokkaa	Poista

<< Edellinen 1 | 2 | 3 | 4 | Seuraava >>

Poista

© 2010

h_ylaosa.php
h_kalenteri_kiinni.php
h_uni_muokkaus.php (kutsuttava sivu)
alaosa.php

Kuva 8.1 Muokkaa univinkkejä -sivu osiin jaettuna

Sivun tulostus aloitetaan sisällyttämällä include-kansiossa sijaitseva h_ylaosa.php-tiedosto tulostettavaan sivuun. Tiedosto sisältää muun muassa sivujen logon ja päämenun. Tämän jälkeen kutsutaan tulosta_valikko()-funktioita, joka luo kunkin osa-alueen oman linkkivalikon. Linkkivalikko on osa h_ylaosa.php-tiedostoa. Seuraavaksi sisällytetään h_kalenteri_kiinni.php-tiedosto, joka sijaitsee myös include-kansiossa. Tiedoston nimi on jäänne pääsivuilta, ja sillä viitataan pääsivuilla olevaan kalenteriin. Ylläpitosivuilla kalenteria ei kuitenkaan ole, mutta tiedostoa tarvitaan lopettamaan linkkivalikon tulostus ja aloittamaan kutsuttavan sivun omat tulostukset kuten kuvasta 8.1 voidaan ha-

vaita. Sivun omat tulostukset luodaan pääosin toiminnallisuuden yhteydessä. Lopuksi sivulle sisällytetään vielä alaosa.php-tiedosto, joka lopettaa sivun tulostuksen. Ylläpitosivuilla ei tarvita omaa alaosa.php-tiedostoa, joten sivuilla voidaan käyttää pääsivujen alaosa.php-tiedostoa, joka sijaitsee pääsivujen juures-
sa.

8.2 Sivun toimintaperiaate

Nyt kun sivurakenne on kuvattu, on sivun toimintaperiaatteen kuvaaminen ja ymmärtäminen helpompaa. Tässä luvussa käytetyt menetelmät koskevat vain Hyvän Olon Päiväkirjan ylläpitosivuja. Luku ei siis ole yleispätevä ohje PHP-sivujen toteutukseen eikä siinä oteta kantaa siihen, mitä muita tapoja on toteuttaa sivuja.

Sivun suoritus tapahtuu ylhäältä alaspäin. Aina, kun sivulle saavutaan tai sivu ladataan painiketta painamalla, koko sivu suoritetaan uudelleen alusta loppuun. Sovellus ei siis hyppää esimerkiksi painikkeen painalluksen johdosta johonkin tiettyyn kohtaan sivua, vaan sivu suoritetaan joka kerta kokonaisuudessaan. Sivun suoritus tapahtuu niin, että vain tietyt toiminnallisuudet suoritetaan tietyin ehdoin, jolloin vältetään turhan koodin suorittaminen.

Kun sivulle saavutaan ensimmäisen kerran, ensimmäisenä alustetaan sivun muuttujiin tarvittavat oletusarvot. Seuraavaksi haetaan tietokannasta ne tiedot, joita halutaan muokata tai näyttää käyttäjälle tai joita muuten tarvitaan sivun toiminnallisuudessa. Kun tiedot on haettu ja käsitelty, voidaan ne sijoittaa tulostusta varten HTML-elementteihin. Sivun päätoiminnallisuus eli tietojen muokkaus ja välitys tapahtuu HTML-lomakkeen eli Form-elementin ja Post-metodin avulla.

Form-elementti on varsinainen lomakealue, johon sijoitetaan tiedot syöttämistä tai näyttämistä varten (Rantala 2002, 96). Form-elementti sisältää tiedon käsittelyyn tarkoitettuja elementtejä kuten tekstikenttiä ja valikoita. Form-elementti voi myös sisältää piilotettuja tietoja, joita ei näytetä käyttäjälle. Lomake suoritetaan painikkeella, jonka tyypiksi on määritetty submit. Kun painiketta on painet-

tu, lähetetään lomakkeen sisältämät tiedot Post-metodin avulla käsiteltäväksi. Käsittely suoritetaan samalla sivulla, jolla lomake on, eli lomake suorittaa itsensä.

Sivu ladataan uudelleen, kun submit-painiketta painetaan. Tällöin lomakkeen tiedot välitetään Post-metodin avulla ikään kuin takaisin sivun alkuun, jossa ne ovat käytettävissä \$_POST-nimisessä taulukossa. Taulukko sisältää elementin nimen, jossa tieto oli, sekä tiedon, jonka elementti sisälsi.

Kun sivun suoritus aloitetaan alusta painiketta painamalla, tapahtuu sivun suoritus samalla tavalla kuin muulloinkin. Tällä kertaa muuttujien alustusten jälkeen suoritetaankin painikkeen painallukselle toteutetut toiminnallisuudet.

Painikkeen painallus havaitaan sivua suoritettaessa if-ehtolauseen ja isset-funktion avulla. If-ehtolause on yksi ohjelmointikielien yleisimmistä rakenteista, jota käytetään tarkistamaan, onko joku ehto tosi tai epätosi. PHP-kielen isset-funktiota puolestaan käytetään tarkistamaan, onko jokin arvo asetettu. Eli jos painiketta on painettu, suoritetaan joku tietty toiminnallisuus, jos ei, jatketaan sivun suorittamista eteenpäin.

Kun painiketta on painettu, suoritetaan toiminnallisuudet, jotka on toteutettu if-ehtolauseen lohkon sisälle. Lohkolla tarkoitetaan aaltosulkeiden sisällä olevaa aluetta. Jos kyseisen painikkeen painalluksen tarkoituksena on esimerkiksi tallentaa lomakkeelle syötetyt tiedot, poimitaan tiedot \$_POST-taulukosta, tarkistetaan ne ja tallennetaan tietokantaan. Tämän jälkeen sivun suoritusta jatketaan normaalisti. Jos tiedot ovat jollain tavalla viallisia, esimerkiksi kokonaisluvuille tarkoitettuihin tekstikenttiin on syötetty kirjaimia, ei tietoja tallenneta tietokantaan. Tällöin käyttäjälle luodaan virheilmoitus ja jatketaan sivun suorittamista eteenpäin.

Kun sivun suorituksessa tullaan jälleen siihen kohtaan, jossa tiedot haetaan tietokannasta, ovat haettavat tiedot nyt päivittyneet painikkeen painalluksesta johtuneen tallennuksen johdosta. Tämä koskee tietenkin vain sellaista tapausta

jossa tietojen tallennus suoritettiin onnistuneesti edellisessä vaiheessa. Muutoin tietokannasta haetaan samat tiedot kuin ensimmäiselläkin keralla.

Lisäksi sivut sisältävät sellaisia toiminnallisuuksia, jotka suoritetaan vain silloin, kun sivulle saavutaan ensimmäisen kerran. Nämä toiminnallisuudet ovat toteutettu if-ehtolauseen, isset-funktion ja \$_POST-taulukon avulla. Kun sivulle saavutaan ensimmäistä kertaa, on \$_POST-taulukko tyhjä. Tarkistamalla, että \$_POST-taulukko on tyhjä, voidaan päätellä, että sivulle saavutaan ensimmäistä kertaa. Tämän tavan käyttö edellyttää, ettei \$_POST-taulukkoa tyhjennetä missään vaiheessa, ellei haluta, että sivun suorittaminen aloitetaan ikään kuin alusta.

8.3 Sivujen sisältö

Kuten jo luvussa 7.3 todettiin, ylläpitosivuille toteutettuja osa-alueita on kahdeksan. Tämä luku sisältää taulukoita (taulukot 8.1 – 8.8), joista kukin esittää yhtä osa-aluetta. Taulukoissa on lueteltu kaikki sivut jotka osa-alue sisältää sekä toiminnallisuudet, joista sivut koostuvat.

Taulukko 8.1 Käyttäjät-osa-alueen sisältö

Käyttäjien hallinta	
Sivu	Toiminto
Lisää käyttäjä (h_kayttaja_lisaa.php)	Lisää käyttäjiä lisättävien listaan.
	Poista valitut käyttäjät lisättävien listasta.
	Poista kaikki käyttäjät lisättävien listasta.
	Tallenna käyttäjät tietokantaan.
Muokkaa Käyttäjiä (h_kayttaja_muokkaus.php)	Lisää käyttäjiä poistettavien listaan.
	Lisää käyttäjä poistettavaksi.
	Poista valitut käyttäjät tietokannasta.
	Lisää käyttäjiä viestin vastaanottajien listaan.
	Vaihda näytettävien käyttäjien lukumäärää.
Lisää käyttäjä muokattavaksi.	
Muokkaa Käyttäjää (h_kayttaja_muokkaa.php)	Tallenna muutokset tietokantaan.
	Peruuta muokkaus.
Muistuta käyttäjiä (h_kayttaja_muistuta.php)	Lähetä viesti käyttäjille.
	Peruuta viestin lähetys.

Taulukosta 8.1 huomataan, että Käyttäjät-osa-alue sisältää perusmuokkaustoimintojen, kuten *lisää*, *poista* ja *muokkaa*, lisäksi paljon muitakin toiminnallisuuksia. Sama koskee kaikkia muitakin osa-alueita. Toiminnallisuuksissa esiintyy usein toimintoja, joilla lisätään jokin tieto listaan tai poistetaan tieto listasta. Listalla tarkoitetaan esikatselua, jossa ylläpitäjällä on vielä mahdollisuus seuloa tietoja ennen kuin lopullinen toiminnallisuus suoritetaan. Esimerkiksi, kun ylläpitäjä haluaa poistaa useita käyttäjiä kerrallaan, hän valitsee poistettavat käyttäjät Muokkaa käyttäjiä -sivulta. Kun halutut käyttäjät on valittu, painetaan Poista-painiketta. Painikkeen painalluksen jälkeen sovellus ei kuitenkaan suoraan poista valittuja käyttäjiä, vaan sovellus näyttää poistettavaksi valitut käyttäjät listalla. Jos listalle on vahingossa valittu käyttäjiä, joita ei haluta poistaa, voidaan ne poistaa listalta, jolloin niitä ei poisteta, kun painetaan Kyllä-painiketta, joka vahvistaa käyttäjien poiston.

Taulukko 8.2 Sisältö-osa-alueen sisältö

Sisällön hallinta	
Sivu	Toiminto
Muuttujien hallinta (h_sisalto_muuttujat.php)	Näytä muokattavat muuttujat.
	Poista valinnat muokattavista muuttujista.
	Tallenna muutokset tietokantaan.
Palauta muuttujat (h_sisalto_palauta.php)	Palauta oletusmuuttujat.
	Palauta viimeksi tallennetut muuttujat.

Sisältö-osa-alue sisältää sovelluksen kannalta erittäin kriittisiä tietoja, jotka vaikuttavat koko sovelluksen toimintaan. Tästä syystä Palauta muuttujat -sivulta löytyy kaksi toiminnallisuutta, joilla voidaan peruttaa tehdyt muutokset (taulukko 8.2). Palauttamalla sovelluksen mukana tulleet oletustiedot, eli oletusmuuttujat, tiedot palautetaan siihen tilaan, jossa ne olivat alun perin. Palauttamalla viimeksi tallennetut muuttujat, tiedot palautetaan siihen tilaan, jossa ne olivat ennen viimeisintä tallennusta.

Taulukko 8.3 Ravinto-osa-alueen sisältö

Ravinto-sivun hallinta	
Sivu	Toiminto
Lisää ruoka-aine (h_ravinto_lisaa.php)	Näytä esikatselu.
	Muokkaa lisättävää ruoka-ainetta.
	Tallenna ruoka-aine tietokantaan.
	Peruuta lisäys.
Muokkaa ruoka-aineita (h_ravinto_muokkaa.php)	Lisää ruoka-aineita poistettavien listaan.
	Lisää ruoka-aine poistettavaksi.
	Poista valitut ruoka-aineet tietokannasta.
	Etsi ruoka-aineita.
	Vaihda näytettävien ruoka-aineiden lukumäärää ja lajitteluperusteita.
Lisää ruoka-aine muokattavaksi.	

Etsi-toiminnon avulla ylläpitäjä pystyy etsimään jotain tiettyä tietoa sovelluksen tietokannasta. Esimerkiksi, jos käyttäjä kirjoittaa Muokkaa ruoka-aineita -sivulla

hakuehdoksi ”an”, hakee sovellus tietokannasta kaikki ruoka-aineet, jotka alkavat kirjaimilla a ja n. Ravinto-osa-alueen lisäksi (taulukko 8.3), etsi-toiminto löytyy lähes kaikilta muiltakin osa-alueilta.

Taulukko 8.4 Liikunta-osa-alueen sisältö

Liikunta-sivun hallinta	
Sivu	Toiminto
Lisää liikuntalaji (h_liikunta_lisaa.php)	Näytä esikatselu.
	Muokkaa lisättävää liikuntalajia.
	Tallenna liikuntalaji tietokantaan.
	Peruuta lisäys.
Muokkaa lajeja (h_liikunta_muokkaus.php)	Lisää liikuntalajeja poistettavien listaan.
	Lisää liikuntalaji poistettavaksi.
	Poista valitut liikuntalajit tietokannasta.
	Etsi liikuntalajeja.
	Vaihda näytettävien liikuntalajien lukumäärää ja lajitteluperusteita.
	Lisää liikuntalaji muokattavaksi.

Monelta sivulta, kuten myös Muokkaa lajeja -sivulta (taulukko 8.4), löytyy toiminto, jonka avulla ylläpitäjä pystyy vaihtamaan sivulla näytettävien tietojen, eli tässä tapauksessa liikuntalajien lukumäärä ja lajitteluperusteita. Oletuksena sovellus näyttää Muokkaa-sivuilla kymmenen tietoa kerrallaan tai niin monta tietoa kuin Muuttujien hallinta -sivulla on määritetty. Muokkaa lajeja -sivulla tiedot on lajiteltu nousevaan aakkosjärjestykseen liikuntalajin nimen mukaan. Muita lajittelu-ehdoja ei oletusarvoisesti ole. Ylläpitäjä voi kuitenkin muuttaa lajitteluperusteita. Ylläpitäjä voi esimerkiksi valita, että sivulla näytetään kerrallaan 20 vain ylläpitäjän lisäämää käyttämätöntä liikuntalajia. Käyttämätön liikuntalaji on sellainen, joka löytyy sovelluksen tietokannasta, mutta sitä ei ole käytetty kertaakaan sovelluksessa. Lajittelua voi vielä tarkentaa, kirjoittamalla Etsi-kenttään jokin hakuehto, jolloin sovellus hakee vain hakuehtoa vastaavat tiedot ja lajittelee ne annettujen lajitteluperusteiden mukaan. Jokainen osa-alue sisältää yksilölliset lajitteluperusteet.

Taulukko 8.5 Uni-osa-alueen sisältö

Uni-sivun hallinta	
Sivu	Toiminto
Lisää univinkki (h_uni_lisaa.php)	Näytä esikatselu.
	Muokkaa lisättävää univinkkiä.
	Tallenna univinkki tietokantaan.
	Peruuta lisäys.
Muokkaa univinkkejä (h_uni_muokkaus.php)	Lisää univinkkejä poistettavien listaan.
	Lisää univinkki poistettavaksi.
	Poista valitut univinkit tietokannasta.
	Etsi univinkkejä.
	Vaihda näytettävien univinkkien lukumäärää ja lajitteluperusteita.
	Lisää univinkki muokattavaksi.

Kaikki osa-alueet joilla on *lisää*- ja *muokkaa*-toiminnot, Käyttäjät-osa-alueetta lukuun ottamatta, käyttävät tietojen lisäämiseen ja muokkaamiseen samaa oman osa-alueensa Lisää-sivua. Esimerkiksi Muokkaa univinkkejä -sivun *muokkaa*-toiminto, eli *lisää univinkki muokattavaksi* -toiminto (taulukko 8.5), on vain linkki Lisää univinkki -sivulle. Linkki sisältää muokattavan tiedon yksilöivän tiedon, jonka avulla sovellus hakee sen tiedot muokattavaksi.

Jos ylläpitäjä ei haluakaan muokata tietoja tai lisätä uutta tietoa, voi hän peruuttaa toimenpiteen painamalla Peruuta-painiketta tai siirtymällä jollekin muulle sivulle. Tällöin tietoja ei tallenneta vaikka tekstikenttiin olisikin tehty muutoksia.

Taulukko 8.6 Stressi-osa-alueen sisältö

Stressi-sivun hallinta	
Sivu	Toiminto
Lisää stressivinkki (h_stressi_lisaa.php)	Näytä esikatselu (kuva tai teksti).
	Muokkaa lisättävää stressivinkkiä.
	Tallenna stressivinkki tietokantaan.
	Palauta alkuperäinen kuva.
	Peruuta lisäys.
Muokkaa stressivinkkejä (h_stressi_muokkaus.php)	Lisää stressivinkkejä poistettavien listaan.
	Lisää stressivinkki poistettavaksi.
	Poista valitut stressivinkit tietokannasta.
	Etsi stressivinkkejä.
	Vaihda näytettävien stressivinkkien lukumäärää ja lajitteluperusteita.
	Lisää stressivinkki muokattavaksi.

Stressivinkit ovat lyhyitä vinkkejä, joita tulostetaan käyttäjille satunnaisessa järjestyksessä, kun he saapuvat pääsivujen Stressisivulle. Vinkki voi olla joko teksti tai kuva. Lisää stressivinkki sivulta löytyy toiminto nimeltä Palauta alkuperäinen kuva (taulukko 8.6). Kun ylläpitäjä haluaa muokata stressivinkkiä joka on kuva, voi hän vaihtaa kuvaksi uuden kuvan ja esikatsella uutta kuvaa. Tällöin esikatselukuva tallentuu väliaikaisesti ylläpitosivujen tmp-kansioon. Jos ylläpitäjä ei ole tyytyväinen uuteen kuvaan, voi hän palauttaa stressivinkin alkuperäisen kuvan *palauta alkuperäinen kuva* -toiminnon avulla. Tällöin esikatselukuva tuhoetaan tmp-kansiosta.

Taulukko 8.7 Tupakka-osa-alueen sisältö

Tupakka-sivun hallinta	
Sivu	Toiminto
Lisää tupakkavinkki (h_tupakka_lisaa.php)	Näytä esikatselu.
	Muokkaa lisättävää tupakkavinkkiä.
	Tallenna tupakkavinkki tietokantaan.
	Peruuta lisäys.
Muokkaa tupakkavinkkejä (h_tupakka_muokkaus.php)	Lisää tupakkavinkkejä poistettavien listaan.
	Lisää tupakkavinkki poistettavaksi.
	Poista valitut tupakkavinkit tietokannasta.
	Etsi tupakkavinkkejä.
	Vaihda näytettävien tupakkavinkkien lukumäärää ja lajitteluperusteita.
	Lisää tupakkavinkki muokattavaksi.

Tupakka osa-alueen hallintasivut ovat ylläpitosivujen yksinkertaisimmat. Kuten taulukosta 8.7 voidaan havaita, sivut eivät sisällä mitään toimintoja, joita ei löydy joiltain muilta sivuilta.

Taulukko 8.8 Tietoiskut-osa-alueen sisältö

Tietoiskujen hallinta	
Sivu	Toiminto
Lisää tietoisku (h_tietoisku_lisaa.php)	Näytä esikatselu.
	Muokkaa lisättävää tietoiskua.
	Tallenna tietoisku tietokantaan.
	Palauta alkuperäinen kuva.
	Peruuta lisäys.
Muokkaa tietoiskuja (h_tietoisku_muokkaus.php)	Lisää tietoiskuja poistettavien listaan.
	Lisää tietoisku poistettavaksi.
	Poista valitut tietoiskut tietokannasta.
	Etsi tietoiskuja.
	Vaihda näytettävien tietoiskujen lukumäärää ja lajitteluperusteita.
	Lisää tietoisku muokattavaksi.

Kuten Lisää stressivinkki -sivu, myös Lisää tietoisuus -sivu sisältää *palauta alkuperäinen kuva* -toiminnon (taulukko 8.8). Toiminto toimii samalla periaatteella kuin Lisää stressivinkki -sivullakin, mutta tietoisuuksien yhteydessä kuvalla tarkoitetaan itse tietoisuuskäsitteen lisäksi lisättävää kuvaa. Lisää tietoisuus -sivulla toiminto sisältää myös mahdollisuuden valita, näytetäänkö kuvaa vai.

Kuten taulukoista voidaan havaita, monet sivut sisältävät samoja toiminnallisuksia. Silti jokaisen osa-alueen sivut ja toiminnallisuudet ovat yksilöllisiä, ja ne on rakennettu yksinomaan kunkin osa-alueen tarpeet huomioiden.

9 KEHITTÄMISMAHDOLLISUUDET

Vaikka Hyvän Olon Päiväkirjan pääsivut ovat olleet käytössä jo joulukuusta 2009 lähtien ja ylläpitosivutkin otetaan pian käyttöön, on sivustossa vielä paljon kehitettävää. Sivuston jatkokehityksen kannalta olisi hyvä, jos sivut muutettaisiin oliopohjaisiksi. Näin uusien toiminnallisuuksien lisääminen ja vanhojen kehittäminen helpottuisi huomattavasti. Lisäksi lähdekoodin tulkitseminen helpottuisi.

Ylläpitosivut ovat nykyään ”putkimaisesti” toteutettu. Tämä tarkoittaa sitä, että ylläpitäjä pääsee muokkaamaan vain ylläpitosivuilla tarkoin rajattua osaa sivuston sisällöstä. Olisi hienoa, jos ylläpitäjä pystyisi muokkaamaan kaikkea sivuilla näkyvää aina teksteistä kuviin ja elementtien sisällöstä valikoiden linkkeihin. Ylläpitosivuille ei välttämättä tarvittaisi omaa erillistä käyttöliittymää, vaan ylläpityökalut voisivat olla osa pääsivuja. Nämä työkalut näkyisivät vain niille käyttäjille, joilla on ylläpito-oikeudet. Esimerkiksi Tietoisuus-sivulla voisi olla jokaisen tietoisuuden yhteydessä Muokkaa-painikke, jolla pääsisi suoraan muokkaamaan kyseistä tietoisuutta. Päivän tavoite kalorijakauman näyttävän kuvan yhteydessä voisi olla painike, jolla pääsisi suoraan muokkaamaan jakauman prosenttiarvoja. Nämä työkalut voisivat olla käytössä myös nykyisten ylläpitosivujen lisäksi. Ne parantaisivat omalta osaltaan sivuston sisällön ylläpidettävyyttä.

Jos pääsivujen käyttäjämäärä tulevaisuudessa kasvaa, täytyy sovelluksen tietokantaa kehittää. Tietokannan rakenne ja toteutus sopii hyvin nykyiseen tarkoi-

tukseensa, mutta käyttäjämäärän kasvaessa se saattaa käydä raskaaksi. Esimerkkinä tästä toimii jokaisen käyttäjän ruokapäiväkirjan tiedot sisältävä taulu. Taulu sisältää yhden rivin jokaista päiväkirjaan tallennettua ravinto-ainetta kohden. Tämä tarkoittaa sitä, että jos käyttäjä lisää päiväkirjaan 20 syötyä ravintoainetta joka päivä kymmenen päivän ajan, kertyy rivejä tuolta ajalta yhteensä 200. Jos käyttäjiä on 100, kertyy rivejä tuolta ajalta 20 000. Tuollaisella rivimäärällä tietokannan käyttö käy hyvin nopeasti raskaaksi ja hitaaksi. Tietokanta on tällä hetkellä suunniteltu noin 50 käyttäjän käyttöön.

Sivuston käyttäjämukavuutta ja vuorovaikutteisuutta voitaisiin hioa käyttämällä PHP:n lisäksi jotain asiakaspuolen tekniikkaa tai kieltä kuten Ajaxia tai JavaScriptiä. Näillä tekniikoilla voitaisiin luoda paljon sellaista, jota ei pelkällä PHP:llä ole mahdollista toteuttaa. Muun muassa sivuston elementeistä saataisiin vuorovaikutteisempia ja monipuolisempia.

Edellä mainittujen kehitysideoiden lisäksi sivuston kehittäjiltä löytyy varmasti runsaasti uusia ideoita sivuston toiminnallisuuden kehittämiseksi.

10 POHDINTA JA YHTEENVETO

Vaikka toteutetussa sovelluksessa jouduttiin tekemään kompromisseja toteutuksen suhteen, kannattaisi mielestäni PHP-sovelluksissa siirtyä käyttämään yhdenmukaisia toteutustapoja. PHP-sovellukset ovat olleet ikään kuin ohjelmoinnin Villi länsi, johtuen mahdollisesti matalasta aloitus kynnyksestä. Internet on pullollaan ”asiantuntijoita”, joista kullakin on oma näkemyksensä siitä, kuinka PHP-sovellukset pitää toteuttaa. Tämä johtaa aloittelijoita helposti harhaan. Jos jo pelkästään siitä kiistellään, kuinka pelkkää tekstiä kannattaa PHP-kielellä tulostaa (echo vastaan print), ollaan mielestäni hakoteillä ja liian kaukana yhdenmukaisuudesta. Toivon, että joku taho ottaa tehtäväkseen PHP-toteutustapojen yhdenmukaistamisen varsinkin isompia sovelluksia tehtäessä. Pelkkien lyhyiden PHP-komentosarjojen kirjoittamisen aika alkaa olla ohi kun siirrytään tekemään olio-pohjaisia PHP-sovelluksia.

Tunnelin päässä näkyy kuitenkin jo valoa. Jatkuvasti tulee toinen toistaan va-
kuuttavampia sovelluskehitysympäristöjä ja -ohjelmia, jotka ovat suunniteltu
pelkästään PHP:lle tai jotka sisältävät tuen PHP:lle ja joilla on yhä helpompi
luoda hallittuja kokonaisuuksia. Näistä mainittakoon muun muassa ilmainen
EllisLabin CodeIgniter, joka on PHP sovelluskehys, ja Microsoftin maksullinen
Expression Web 3, joka on web-sovellusten tekoon tarkoitettu työkalu.

Ohjelmien toteuttaminen pelkällä tekstieditorilla alkaa olla jo mennyttä aikaa.
Toki pienempiä PHP-sovelluksia voidaan yhä toteuttaa tekstieditoreilla, mutta
isompien sovellusten toteutukseen kannattaa käyttää muita työkaluja. Ohjelman
luominen pelkällä tekstieditorilla on mielestäni kuitenkin sellainen taito, jonka
jokaisen ohjelmoijan soisi osaavan. Kun kirjoittaa ohjelman alusta loppuun itse,
tietää tasan tarkkaan, miten ohjelma toimii ja mitä kullakin lähdekoodin rivillä
tehdään. Samalla oppii käyttämään kieltä perusteellisesti. Nykyään on myös
olemassa sellaisiakin työkaluja, joilla voidaan luoda yksinkertaisia ohjelmia kir-
joittamatta riviäkään koodia. Jos näitä ohjelmia kuitenkin halutaan kustomoida,
on edes jonkinlaisten ohjelmointi taitojen omaaminen välttämätöntä.

Mielestäni yksi suurimmista ongelmista web-sovellusten kehityksessä on web-
selainten väliset erot. Ongelma ei suoranaisesti liity PHP-kieleen, vaan kaikkiin
CSS-tyylejä käyttäviin sovelluksiin ja sivuihin. Usein on tehtävä kompromisseja
ulkoasun kohdalla, koska sivua ei saa millään näyttämään samalta kahdella eri
selaimella, käyttämättä eri selaimille tarkoitettuja tyyliäluokkia tai -tiedostoja.

Hyvänä esimerkkinä voidaan käyttää nykyään suosittavia div-elementtejä, jotka
ovat korvaamassa perinteiset taulukot sivuja rakennettaessa. Div-elementtien
tarkoituksena on helpottaa sivujen rakentamista ja niiden muokkaamista, mutta
samalla ne myös saattavat teettää lisätöitä. Eri selaimia varten on usein kirjoj-
tettava eri luokkia ja deklaraatioita, jotta sivut näyttäisivät samalta eri selaimis-
sa. Web-sovellusten ulkoasujen ja käyttöliittymien toteuttaminen helpottuisi
huomattavasti jos web-selainten välisiä eroavaisuuksia ei olisi.

Kaikesta huolimatta työn tavoitteet saavutettiin ainakin siinä määrin, että työn
tuloksena saatiin toimivat ylläpitosivut Hyvän Olon Päiväkirjalle. Koska ylläpi-

tosivujen käyttöönottoa ei vielä ole suoritettu, on vaikea sanoa, vastaavatko työn tulokset asiakkaan toiveita. Mutta koska toteutin myös sovelluksen pääsivut, oli minulla erittäin vahva näkemys siitä, mitä ylläpitosivujen pitää sisältää, jotta pääsivujen sisällön hallinnointi niillä onnistuisi. Noiden näkemysten perusteella syntyivät suunnitelmat, jotka lopulta hyväksyttiin asiakkaalla. Tästä syystä ylläpitosivujen sisällön puolesta tavoitteet ainakin täyttyivät. Omissa testeissäni ylläpitosivujen käyttö on osoittautunut erittäin vaivattomaksi. Nähtäväksi kuitenkin jää, mitä mieltä asiakas asiasta on.

Koska kyseessä oli sovellus, jota olen ollut suunnittelemassa ja toteuttamassa alusta lähtien, oli aihe minulle erittäin läheinen. Oli mielenkiintoista tarkastella eri toteutustapoja ja tekniikoita. Kun tekee paljon töitä samalla tekniikalla, on helppo juurtua käyttämään samoja tapoja ja toimintamalleja työstä toiseen. Näin tulee myös toistettua ne samat puutteet ja virheet, joita omat toteutustavat saattavat sisältää. Samaa kaavaa noudattamalla ei myöskään tapahdu kehitystä. Välillä on hyvä tutkailla uusia näkökulmia ja näin löytää parempia tapoja töiden toteuttamiseen. Mielestäni nämä ylläpitosivut ovat selkein PHP:llä tekemäni kokonaisuus, vaikkei niiden toteutustapa aivan noudattanutkaan niitä suunnitelmia, joita minulla aluksi oli.

KUVAT

Kuva 3.1 Saimaan ammattikorkeakoulun organisaatiokaavio (Saimaan ammattikorkeakoulu), s. 11

Kuva 4.1 Vesiputousmalli (Oulun kauppaoppilaitos), s. 12

Kuva 7.1 Muokkaa liikuntalajeja -sivu, s. 28

Kuva 8.1 Muokkaa univinkkejä -sivu osiin jaettuna, s. 33

TAULUKOT

Taulukko 8.1 Käyttäjät-osa-alueen sisältö, s. 37

Taulukko 8.2 Sisältö-osa-alueen sisältö, s. 38

Taulukko 8.3 Ravinto-osa-alueen sisältö, s. 38

Taulukko 8.4 Liikunta-osa-alueen sisältö, s. 39

Taulukko 8.5 Uni-osa-alueen sisältö, s. 40

Taulukko 8.6 Stressi-osa-alueen sisältö, s. 41

Taulukko 8.7 Tupakka-osa-alueen sisältö, s. 42

Taulukko 8.8 Tietoiskut-osa-alueen sisältö, s. 42

LÄHTEET

Boumphrey, F. & Greer, C. & Raggett, D. & Raggett, J. & Schnitzenbaumer, S. & Wugofski, T. 2000. XHTML Ohjelmoijan käsikirja. Helsinki: Oy Edita Ab.

Etelä-Karjalan koulutuskuntayhtymä.

<http://www.ekky.fi/>

(Luettu: 12.5.2010)

Heinisuo, R. & Rauta, I. 2007. PHP ja MySQL – Tietokantapohjaiset verkkopalvelut. 4. uudistettu painos. Helsinki: Talentum Media Oy.

Hietanen, P. 2004. C++ ja olio-ohjelmointi. Jyväskylä: Docendo Finland Oy.

Huhtamäki, J. 2005. 7307010 Hypermedian ohjelmointi, luentorunko. 10 Kohti ketterää ohjelmistokehitystä.

<http://matwww.ee.tut.fi/hmopetus/hm-ohj/2005/pruju/hmohj05-132-143.pdf>

(Luettu: 6.5.2010)

KK Mediat. Artikkelipankki. Johdatus tyylikieliin ja CSS:n historiaan.

<http://www.2kmediat.com/css/johdanto.asp>

(Luettu: 12.5.2010)

Kontio, M., Niskanen, P. & Virimaa, K. 2001. WWW-ohjelmointi Trainer Kit. 2. painos. Helsinki: Oy Edita Ab.

Korpela, J.K. 2003. CSS-tyylit. Jyväskylä: Docendo Finland Oy.

Korpela, J.K. 2000. RTF-tiedostomuoto.

<http://www.cs.tut.fi/~jkorpela/rtf/>

(Luettu: 8.5.2010)

Korpela, J.K. 1998. CSS-termejä suomeksi ja englanniksi.

<http://www.cs.tut.fi/~jkorpela/styles/termit.html>

(Luettu: 7.5.2010)

Lehtonen, K. 2001. HTML-opas.

<http://cs.stadia.fi/~lehtk/html.html>

(Luettu: 12.5.2010)

Morgan, B., Perkins, J., Plew, R. R. & Stephens, R. K. 2001. SQL-tietokantaohjelmointi - Trainer Kit. Helsinki: Oy Edita Ab.

Peltomäki, J. 2001. JavaScript. Jyväskylä: Docendo Finland Oy.

Oulun kauppaoppilaitos. Johdatus tietojärjestelmiin. 2 Kehittämistyön vaiheet ja elinkaarimallit.

http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/kehittamistyon_vaiheet_ja_elikaarimallit/kehittamistyon_vaiheet_ja_elinkaarimallit_asia.htm

(Luettu: 9.5.2010)

Oulun kauppaoppilaitos. Vesiputousmalli.

http://www.okol.org/verkkokurssit/datanomi/tietojarjestelmien_kaytto_ja_kehittaminen/johdatus_tietojarjestelmiin/images/kuva2_2.jpg

(Luettu: 9.5.2010)

Oulun seudun ammattikorkeakoulu. Open source. MySQL opetusmateriaali. Transaktio.

<http://www.ratol.fi/opensource/mysql/transaktio.htm>

(Luettu: 7.5.2010)

Rantala, A. 2002. PHP - web-ohjelmoinnin peruskirja. Jyväskylä: Docendo Finland Oy.

Saimaan ammattikorkeakoulu. Saimaan amk lyhyesti.

<http://www.saimia.fi/fi-FI/tietoja-saimaan-amk/saimaan-amk-lyhyesti>

(Luettu: 12.5.2010)

Saimaan ammattikorkeakoulu. Organisaatio.

http://www.saimia.fi/fi-FI/images/saimaan_amk_organisaatio.jpg

(Luettu: 12.5.2010)

Sivut.web. HTML-opas: Yleistä.

<http://www.sivut.org/html/oppaat/yleista.php>

(Luettu: 12.5.2010)

Tietokone. 2008. Scite - Scintilla Text Editor - Ohjelmoijan tekstieditori Windowsiin.

http://www.tietokone.fi/softa/windows/scite_scintilla_text_editor

(Luettu: 12.5.2010)

Tietotekniikan liitto ry. 2003. ATK-sanakirja. 12. uudistettu paino. Helsinki: Talentum Media Oy.

Tietotekniikan liitto ry. ATK-sanakirja.

<http://www.ttlry.fi/atk-sanakirja/su024.htm>

(Luettu: 7.5.2010)

Turun yliopisto, Informaatioteknologian laitos. 2003. Ohjelmistoprosessi.

[http://staff.cs.utu.fi/opinnot/kurssit/ohjelmistotuotanto/syksy_2003/luennot/Ohjelmistoprosessi\(2\).pdf](http://staff.cs.utu.fi/opinnot/kurssit/ohjelmistotuotanto/syksy_2003/luennot/Ohjelmistoprosessi(2).pdf)

(Luettu: 9.5.2010)

Zandstra, M. 2001. PHP Trainer Kit. Helsinki: Oy Edita Ab.

```

<?php
require_once('muuttujat.php');
require_once('funktiot.php');
//*****
// SIVUN OMAT MUUTTUJAT
//*****
$muuttuja = "Terve! ";
$tulostus = "";
//*****
// SIVUN TOIMINNALLISUUS
//*****
//*****
// BTN_PAINIKE
//*****
if(isset($_POST['btn_painike'])){
    $muuttuja .= "Painikkeen painalluksen jälkeen.";
}
//*****
// PAINIKKEISTA RIIPPUMATTOMAT TOIMENPITEET
//*****
// suoritetaan vain kun sivulle saavutaan ensimmäistä kertaa
if(!$_POST){
    $muuttuja = "Ensimmäistä kertaa.";
}
// suoritetaan joka kerta
$tulostus = "<p>".$muuttuja."<p>";
//*****
// SIVUN TULOSTUS ALKAA
//*****
require_once('include/h_ylaosa.php');
//*****
// OMA LINKKIVALIKKO
//*****
tulosta_valikko($valikko_linkit);

require_once('include/h_kalenteri_kiinni.php');
//*****
// SIVUN OMAN SISÄLLÖN TULOSTUS
//*****
echo $tulostus;

echo "
<form action=\"".$_SERVER['REQUEST_URI']."\" method=\"post\">
    <input type=\"submit\" name=\"btn_painike\" value=\"OK\" />
</form>
";

require_once('../alaosa.php');
?>

```

[Pääsivut](#)

Saimaan ammattikorkeakoulu
Saimaa University of Applied Sciences

Etusivu Käyttäjät Sisältö Ravinto Liikunta Uni Stressi Tupakka Tietoiskut Ohje [Kirjaudu ulos](#)

> Pääsivu
> **Lisää liikuntalaji**
> Muokkaa lajeja

Hallinta: Muokkaa liikuntalajia/aktiiviteettiä (Torstai 20.05.2010) [\(Ohje\)](#)

Lajiaktiiviteetti:

Syötä kulutus kiloa kohden minuutissa:

Kulutus: kcal/kg/min

Tai laske kulutus seuraavien tietojen perusteella:

Suorituksen kesto: min
Paino: kg
Kulutus: kcal

Valitse mihin kohtaan liikuntajärjestelmässä laji sijoittuu:

Kestävyyyskunto: reippaasti
 rasittavasti

Lihaskuntoa ja liikehallintaa: Ei
 Kyllä

© 2010

[Pääsivut](#)

Saimaan ammattikorkeakoulu
Saimaa University of Applied Sciences

Etusivu Käyttäjät **Sisältö** Ravinto Liikunta Uni Stressi Tupakka Tietoiskut Ohje [Kirjaudu ulos](#)

> Pääsivu
> Muuttujien hallinta
> **Palauta muuttajat**

Hallinta: Muuttujien palautus (Perjantai 21.05.2010) [\(Ohje\)](#)

Tällä sivulla voit palauttaa alkuperäiset muuttajat tai ennen viimeisintä tallennusta voimassa olleet muuttajat. Palautus tarkoittaa sitä, että palautettavat muuttajat otetaan käyttöön ja edelliset poistetaan käytöstä.

Palauta alkuperäiset muuttajat

Alkuperäiset muuttajat ovat ne muuttajat, jotka olivat voimassa kun sivut otettiin käyttöön. Kun palautat alkuperäiset muuttajat, eivät viimeeksi voimassa olleet muuttajat ole enää palautettavissa

Palauta edelliset muuttajat

Edelliset muuttajat ovat ne muuttajat, jotka olivat voimassa ennen viimeisintä tallennusta. Kun palautat edelliset muuttajat, eivät viimeeksi voimassa olleet muuttajat ole enää palautettavissa

© 2010