

Asta Haapalainen

KESKIKOSKEN LOMAMÖKKIEN INTERNET-SIVUT

Insinöörityö
Kajaanin ammattikorkeakoulu
Tekniikan ja liikenteen ala
Tietotekniikan koulutusohjelma
Kevät 2005



Osasto Tekniikka	Koulutusohjelma Tietotekniikka
Tekijä(t) Asta Haapalainen	
Työn nimi Keskikosken Lomamökkien Internet-sivut	
Vaihtoehtoiset ammattiopinnot Ohjelmistotekniikka	Ohjaaja(t) Raili Simanainen
Aika 24.01.2005	Sivumäärä 45
Tiivistelmä <p>Insinööriyön tavoitteena oli suunnitella ja toteuttaa Keskikosken Lomamökeille toimivat Internet-sivut.</p> <p>Keskikosken Lomamökit vuokraa lomamökkejä. Yritys on perustettu tammikuussa 2004, ja tällä hetkellä toimipaikka on Sonkakoskella. Yritys on perheyrittys. Yrityksen liiketoiminnan kannalta oli tarpeellista kehittää Internet-sivut.</p> <p>Internet-sivujen suunnittelussa on käytetty HTML-, CSS-, JAVA-, PHP- ja SQL-ohjelmointikieltä. Tietokanta toteutettiin MySQL -tietokantana, jonka sisältöä hallitaan WWW-selaimella PHP:llä kirjoitettujen palvelinsivujen avulla.</p> <p>Insinööriyön tuloksena saatiin valmis tietokanta ja Keskikosken Lomamökkien Internet-sivut. Internet-sivut ovat olleet käytössä 1.1.2005 alkaen. Tähän asti ne ovat toimineet moitteettomasti.</p>	
Luottamuksellinen Kyllä Ei <input checked="" type="checkbox"/> x	
Hakusanat	
Säilytyspaikka	



Faculty Engineering	Degree programme Information Technology
Author(s) Asta Haapalainen	
Title Internet Pages of Keskkikoski Holiday Cottages	
Optional professional studies Software Engineering	Instructor(s) / Supervisor(s) Raili Simanainen
Date 24 January 2005	Total number of pages 45
Abstract <p>The purpose of this Bachelor's thesis was to design and implement the Internet pages for the Keskkikoski Holiday Cottages company.</p> <p>Keskkikoski Holiday Cottages rents holiday cottages. The company was established in January 2004. At the moment it has an office in Sonkakoski. The company is a family-owned. It was necessary to develop the Internet pages for the business of the company.</p> <p>The Internet pages were designed by the programming languages HTML, CSS, JAVA, PHP and SQL. The database consists of the MySQL database and WWW documents written in PHP. The WWW documents are used to manage the database.</p> <p>The final outcome of this thesis was a complete database and the Internet pages of Keskkikoski Holiday Cottages. The Internet pages were introduced in January 2005. So far they have worked well.</p>	
Confidential Yes No <input checked="" type="checkbox"/>	
Keywords	
Deposited at	

ALKUSANAT

Tämä insinööryö on tehty Kajaanin ammattikorkeakoulussa Sonkakoskella toimivan Keskikosken Lomamökkien tilauksesta.

Haluan kiittää insinööryön valvojaa opettaja Raili Simanaista Kajaanin ammattikorkeakoulusta ja yrittäjä Pertti Karista Keskikosken Lomamökeistä. Lisäksi haluan kiittää kieliasun tarkastamisesta lehtori Eero Soinista ja yliopettaja Kaisu Korhosta.

Kajaanissa 23.1.2005

Asta Haapalainen

SISÄLLYSLUETTELO

Tiivistelmä

Abstract

Alkusanat

Lyhenteet, käsitteet ja määritelmät

1 JOHDANTO	7
2 TIETOKANTAPOHJAINEN SOVELLUS INTERNETISSÄ	8
2.1 UML-mallinnuskieli	8
2.2 Tietokannan suunnittelu	9
2.3 Kolmikerrosmalli	11
2.4 MySQL-tietokantapalvelimena.....	12
2.5 Ohjelmointi	13
3 KESKIKOSKEN LOMAMÖKKIEN INTERNET-SIVUJEN VAATIMUSMÄÄRITTELY.....	16
3.1 Yleiskuvaus	16
3.2 Ongelmat ja rajoitukset.....	18
3.3 Toiminnalliset vaatimukset	18
3.5 Ei-toiminnalliset vaatimukset	23
3.6 Muut vaatimukset	23
3.7 Käyttöliittymät	24
3.8 Tiedot ja tietokanta	24
4 KESKIKOSKEN LOMAMÖKKIEN INTERNET-SIVUJEN SUUNNITTELU	26
5 KESKIKOSKEN LOMAMÖKKIEN INTERNET-SIVUJEN TOTEUTUS	27
5.1 Tekninen ratkaisu	27
5.2 Sivupohjan toteutus	27
5.3 Lomakkeen toteutus	32
5.4 Tietokannan toteutus	34
5.5 Yhteyden luominen tietokantaan	35
5.6 Tietokantaan tallentaminen	37
5.7 Ajastetun tekstikentän toteutus.....	40
5.8 Sivujen testaus	42
5.9 Sivujen ylläpito ja käyttöönotto	43
6 YHTEENVETO	44
LÄHDELUETTELO	45

Lyhenteet, käsitteet ja määritelmät

CSS (Cascading Style Sheets)

Yksinkertainen tyylikieli, jolla voidaan liittää tyylimäärytykset HTML-elementteihin.

HTML (HyperText Markup Language)

Sillä kuvataan WWW-palvelimesta tekstin hakeneelle asiakasohjelmalle tekstin rakenne eli tekstin osien tehtävät.

Kolmikerrosmalli (*Three -Tier model*)

Kolmeen kerrokseen perustuva sovellusmalli. Toteutustapa WWW-sovelluksissa yleensä asiakaskerros - esitys, välikerros - käsittely ja pohjakerros - tietovarasto.

MySQL

Suosituin avoimeen lähdekoodiin perustuva reaalitytokantajärjestelmä. Suorituskykyinen, joustava ja monipuolinen varsinkin WWW-sovelluksissa.

Palvelinsivu

WWW-dokumentti, johon on HTML-merkinnän sekaan upotettu palvelimessa suoritettavia skriptejä erikoiselementtien sisälle. Puhutaan myös upotetuista tekniikoista.

PHP (*Hypertext Preprocessor*)

Useilla alustoilla toimiva avoin upotus- tai palvelinsivutekniikka. Käytettävä skriptikielen syntaksi on lainattu pääosin C:stä.

Skripti

Yleisesti käytetty nimitys WWW-palvelimessa tulkattavasta ohjelmasta, jolloin WWW-palvelin on laajennettu käytetyn skriptikielen tulkilla.

1 JOHDANTO

Insinööriyön tavoitteena on suunnitella ja luoda Internet-sivut Keskikosken Lomamökkien käyttöön. Sivujen tarkoituksena on toimia Keskikosken Lomamökkien esittely- ja mainossivuina.

Keskikosken Lomamökit on tammikuussa 2004 perustettu perheyritys, joka välittää vuokralle ensiluokkaisia ja erittäin tasokkaita loma-asuntoja. Yrityksen toimipiste on tällä hetkellä Sonkakoskella. Myös loma-asunnot sijaitsevat Sonkakoskella kauniissa suvantomaisemassa.

Keskikosken Lomamökeillä ei ole olemassa entuudestaan Internet-sivuja, joten toteuttaminen lähtee liikkeelle yrittäjän ja insinööriyöntekijän visioiden yhdistämisestä. Sivujen tulisi olla yritystä kuvaavat ja yksinkertaisen selkeät. Niiden päivittäminen tulisi myös olla mahdollisimman yksinkertaista ja helppoa.

Vaatimusten perusteella päädyttiin muuttumattomaan sivupohjaan ja muuttuvaan sisältöosioon WWW-dokumentissa. Sivuille liitetään myös yhteyslomake, jolla sivuilla kävijä voi lähettää kyselyn yrittäjälle. Yhteyslomakkeesta lähetetty tieto tallentuu palvelimella sijaitsevaan tietokantaan, josta yrittäjä hakee tiedon ja vastaa sähköpostitse kyselyn lähettäjälle. Sivujen laadinnassa käytetään HTML-, CSS-, PHP- ja JavaScript-koodeja. Tietokantana toimii palvelimella oleva MySQL- tietokanta, johon luodaan toiminnot SQL-kielellä.

2 TIETOKANTAPOHJAINEN SOVELLUS INTERNETISSÄ

2.1 UML-mallinnuskieli

Mallinnuskieli on kuvauskieli eli notaatio, joita menetelmissä käytetään suunnitelmien kuvaamiseen. Prosessi puolestaan on menetelmissä käytettävä ohjeistus suunnitelman laatimiseen tarvittavista vaiheista. [1, s. 2.]

UML sisältää yhdeksän erityyppistä kaavioesitystä, joilla kuvataan järjestelmän korkean tason toiminnallisuutta (*käyttötapauskaaviot*), järjestelmän staattista ja dynaamista rakennetta (*oliokaaviot*, *luokkakaaviot*, *sijoittelukaaviot*, *komponenttikaaviot*) sekä järjestelmän dynaamista käyttäytymistä (*sekvenssikaaviot*, *yhteistyökaaviot*, *tilakaaviot*, *aktiiviteettikaaviot*). Nämä eri kaavioesitykset ovat kuitenkin osittain päällekkäisiä.

Eri kaaviotyyppjä voidaan menetelmästä riippuen soveltaa eri ohjelmistokehityksen vaiheissa. Esimerkiksi käyttötapauskaaviot ovat luontevia analyysivaiheessa. [2, s. 125.]

Käyttötapauskaavio (*use case diagram*) kuvaa järjestelmän käyttötapausten väliset suhteet ja käyttötapauksiin osallistuvat järjestelmän ulkoiset toimijat. Käyttötapauskaaviota käytetään järjestelmän käyttöyhteyden ja vaatimusten korkean tason kuvaukseen. Käyttötapaus (*use care*) on kuvaus järjestelmän ja sen käyttäjien välisestä vuorovaikutuksesta tietyn, käyttäjän kannalta mielekkään tuloksen aikaansaamiseksi. Insinööriyön mukainen malli käyttötapauskaaviosta kohdassa 3.3 [2, s. 127.]

2.2 Tietokannan suunnittelu

Tietokanta on kokoelma yhteen liittyvää tietoa. Tiedolla tarkoitetaan tosiasioita, jotka voidaan kirjata ja joilla on jokin merkitys. Tietokanta voi olla hyvinkin pieni ja yksinkertainen tai vastaavasti erittäin suuri ja monimutkainen. Tässä työssä käytettävä tietokanta on pienehkö.

Tietokannan suunnittelussa on olemassa yhtä monta erilaista tapaa kuin on suunnittelijaakin. Suunnittelun eri vaiheita voivat olla esimerkiksi:

- **Vaatimusten määrittelyssä ja analyysissä** selvitetään järjestelmältä vaadittavat ominaisuudet. Apuna käytetään usein käyttötapaustekniikkaa.
- **Käsitteellisessä mallintamisessa** laaditaan käsitekaava, jolla kuvataan tietokannan sisältö ja rakenne. Kaava tehdään joko käsiteanalyysinä tai oliomallinnuksena.
- **Tehtävä- ja transaktiosuunnittelussa** mallinnetaan tehtävät ja transaktiot, joilla tietokannan tietoja käytetään.
- **Käyttöliittymän suunnittelussa** mallinnetaan sovelluksen rakenne ja sovellukseen kuuluvien näyttöjen sisältö ja rakenne.
- **Tietokannan loogisessa suunnitteluvaiheessa** käsitekaavan pohjalta laaditaan sisällöstä ja rakenteesta relaatiokaava käytettävissä olevalla DDL:llä (*Data Definition Language*).
- **Komponenttien suunnitteluvaiheessa** kiinnitetään käytettävä arkkitehtuuri, muodostetaan komponentteja ja määritellään ne.
- **Fyysisessä suunnitteluvaiheessa** päätetään tietokannan tallennusrakenteista ja saantimenetelmistä. Tarkoituksena on tuottaa suorituskyvyn ja muistitilan käytön kannalta optimaalinen fyysinen rakenne tietokanavalle.
- **Toteutusvaiheessa** toteutetaan edellisten osien suunnitelmat ohjelmointikielillä ja testataan toteutuksen toimivuus. [3, s. 18.]

Käsitteellinen mallintaminen

Käsitteelliseen mallintamiseen on useita eri menetelmiä. Tunnetuin ja käytetyin on Chenin vuonna 1976 esittelemä ER-malli (*Entity-Relationship model*).

ER-malli koostuu useasta erilaisesta osasesta, joita kuvataan tietyillä kuvioilla, jotka yhdistetään toisiinsa tietyillä suhteilla. Näistä muokataan ER-diagrammi, josta nähdään kaikki tarvittavat asiat ja niiden suhteet sekä toimintatavat toisiinsa nähden.

Kohteella (*entity*) tarkoitetaan tunnistettavissa olevaa asiaa tai tapahtumaa (*a thing that can be distinctly identified*). Heikoksi kohteeksi (*weak entity*) kutsutaan kohdetta, jonka olemassaolo riippuu toisesta kohteesta. Se ei siis voi olla olemassa, jos tätä toista kohdetta ei ole olemassa.

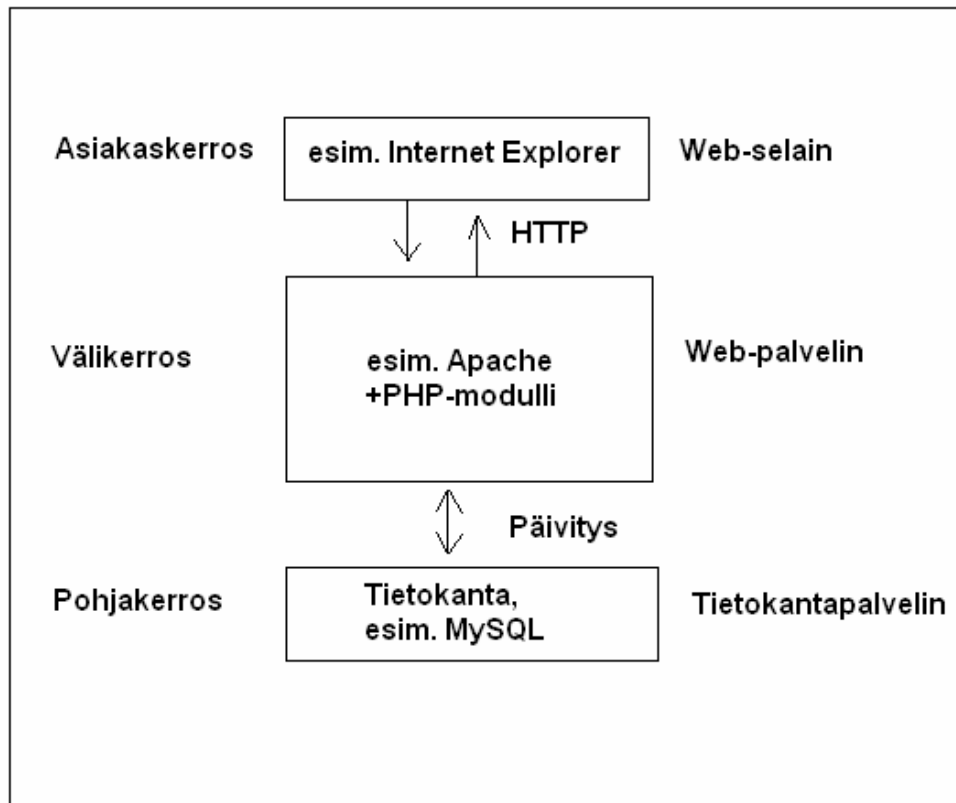
Jokaisella samantyyppisellä kohteella on tiettyjä yhteisiä ominaisuuksia (*properties*). Ominaisuudet voivat olla yksi- tai moniarvoisia, ja ne voivat saada tyhjän arvon tai tuntemattoman arvon. Ominaisuuden arvo voi olla johdettu muista tiedoista.

Suhteella (*relationship*) tarkoitetaan vähintään kahden kohteen välillä vallitsevaa riippuvuutta. Suhteen aste (*degree*) määräytyy suhteeseen liittyvien kohteiden lukumäärän mukaan. Kardinaalisuus (*cardinality*) kertoo, kuinka moneen suhteeseen kohde voi samaan aikaan osallistua ja kuinka monta kohdetta voi samaan aikaan osallistua tiettyyn suhteeseen.

Jo olemassa oleville kohteille voidaan määritellä alityyppejä (*subtypes*), jotka perivät (*inherit*) kaikki ominaisuudet ylityypeiltään. Jokainen kohde on vähintään yhtä kohdetyyppiä mutta voi olla samaan aikaan useampaakin. Insinööriön mukainen malli ER-kaaviosta on kohdassa 3.8 [3, s. 18 - 20]

2.3 Kolmikerrosmalli

Alla olevassa esimerkissä on kuvattu kolmikerrosmallin toiminta periaate käytännössä (kuva 1).



Kuva 1. Kuvaus kolmikerrosmallista

Pohjakerros

Kolmikerroksisessa WWW-sovelluksessa pohjakerros toimii tyypillisesti tiedon varastointikerroksena. Toteutustapana on yleensä tietokantapalvelimessa sijaitseva tietokanta. Pohjakerros toimii palvelimen välikerroksena. Jos pohjakerroksena on relaatiotietokanta, se ottaa vastaan tietokantakomennot ja suorittaa niiden perusteella tarvittavat toiminnot. Pohjakerros huolehtii myös siitä, että tietokantaa voi käyttää useampi käyttäjä yhtä aikaa. Pohjakerros tarjoaa myös ominaisuuksia tietoturvan, käyttöoikeuksien ja tiedon eheyden hallintaan. [4, s. 8]

Välikerros

WWW-sovelluksessa välikerros muodostuu ajoympäristöstä ja WWW-palvelimesta. Välikerros ottaa vastaan asiakaskerroksen pyynnöt, käsittelee ne ja luo niiden perusteella pyyntönsä tietokantapalvelimelle. Sen jälkeen välikerros käsittelee pohjakerrokselta saamansa vastaustiedot ja luo HTML-dokumentin, jonka WWW-palvelin lähettää asiakaskerrokselle. Välikerros sisältää myös skriptejä, palvelinsivuja ja palvelinkomponentteja. [5, s. 8]

Asiakaskerros

WWW-sovelluksessa käyttäjän WWW-selain muodostaa asiakaskerroksen. Se toimii käyttöliittymänä, joka muodostuu vähintään HTML-elementistä ja selaimen ominaisuuksista. Asiakaskerros vastaa tiedon esittämisestä käyttäjälle sen jälkeen kun se on noudettu pohjakerroksesta ja muokattu välikerroksessa. [5, s. 9]

2.4 MySQL-tietokantapalvelimena

MySQL on monipuolinen, joustava ja suorituskykyinen reaalitytökanta, jota käytetään niin suurten kuin pientenkin WWW -palvelujen taustalla. MySQL noudattaa asiakas-palvelin-arkkitehtuuria, jossa sovellukset eivät koskaan käsittele tietokantaa suoraan vaan käsittely tapahtuu aina palvelinohjelman kautta. MySQL-palvelimeen voidaan ottaa yhteys ainakin ODBC- tai Java/JDBC-, PHP-, Perl-, Python- ja Tcl-tekniikoin, tai sitten sen oma C-oliopohjainen C++ -ohjelmointirajapinta on käytettävissä. MySQL on ruotsalaisen T.c.X Ab:n kehittämä.

MySQL:n rakenne, koodin tehokas uudelleenkäyttö ohjelmassa ja minimalistinen lähestymistapa monipuolisten toimintojen toteuttamisessa tekevät siitä nopean, stabiilin ja helposti muunneltavan. [6, s. 33]

2.5 Ohjelmointi

HTML

HTML:n varsinainen tehtävä on kuvata tekstin rakenne eli tekstin osien tehtävät, ei niinkään niiden ulkoasua. HTML:n avulla kuvataan WWW-palvelimesta tekstin hakeneelle asiakasohjelmalle tekstin rakenne. Eri WWW-asiakasohjelmistot näyttävät HTML-tekstit eri tavalla, joten laajempaan käyttöön tarkoitetut tekstit kannattaa testata useilla eri asiakasohjelmilla.

HTML tarjoaa myös tavan esittää kuvia, ääntä ja animaatioita tekstin seassa. HTML:ää voi kirjoittaa miltei millä tahansa tekstieditorilla. Skandinaaviset merkit näkyvät WWW-sivuilla oikein, jos tekstiä kirjoitettaessa on käytössä ollut ISO -Latin 1 -merkistö. HTML-kielen rakenne on hierarkkinen, mikä helpottaa koodin hahmottamista. [7]

CSS

CSS on yksinkertainen tyylikieli, jolla voidaan liittää tyylimääritykset HTML-elementteihin. Kullekin elementtityypille voidaan määritellä yksilöllinen tyyli, joka kuvaa esimerkiksi fontit, värit, marginaalit ja sijainnit. Tyylimäärittelyjen käyttäminen antaa tekijöille lisää mahdollisuuksia sivujen ulkoasun määrittelyille. Määrittelyt voidaan liittää HTML-dokumenttiin tai esittää omassa tiedostossa, jolloin useampi dokumentti voi käyttää samoja määrittelyjä. Näin mahdollistetaan yhtenäinen ulkoasu ja helppo päivitettävyys.

CSS perustuu sääntöihin ja tyylimäärittelyihin. Sääntö on yhden ja useamman elementin tyylimäärittely. HTML-dokumentin muotoilussa tyylimäärittely muodostuu yhdestä tai useammasta säännöstä, joita selain käyttää.

Sääntö muodostuu kahdesta osasta, valitsimesta ja määrittelyosasta. Valitsin on linkki HTML-dokumentin ja tyylin välillä. Se määrittää, mitä elementtiä/elementtejä määrittely koskee. Määrittelyosa ilmaisee, millainen vaikutus säännöllä on. Määrittelyosa jakautuu ominaisuuteen ja arvoon.

Säännöt voidaan kirjoittaa tiivistetyssä muodossa. Tämä nopeuttaa kirjoitusvaihetta sekä lyhentää hieman latausaikaa. Seuraavassa esimerkki-
taulukko, tyylimäärittelymalli (taulukko 6).

Taulukko 1. Tyylimäärittely malli

```
H1 { color : red}
```

Kyseessä on tyyppivalitsin, jolla määritellään kaikki H1-tasoiset otsikot esitettäväksi punaisella. Esimerkissä *H1* on valitsin, *color* ominaisuus ja *red* arvo. [8]

PHP

Rasmus Lerdorf aloitti PHP:n (*Hypertext Preprocessor*) kehitystyön vuonna 1994. PHP on työväline dynaamisten Web-dokumenttien luomiseen. Laajasti ymmärrettynä PHP viittaa sekä varsinaiseen PHP-kielen että niihin teknisiin ratkaisuihin, joilla PHP-kielisten ohjelmien suorittaminen on mahdollista. Kokonaisuutena voidaan puhua PH-ympäristöstä. PHP on suunniteltu erityisesti Web-sovelluskehitystä varten. Tämän vuoksi PHP mahdollistaa ammattilaisille Web-sovellusten nopean kehittämisen ja vasta-alkajille matalan aloituskynnyksen.

PHP on HTML-dokumenttien sisään upotettava Web-palvelimella tulkettava ohjelmointikieli. Tulkattavia ohjelmointikieliä kutsutaan yleisesti skriptikieliksi. PHP:n syntaksi on lainattu suurimaksi osaksi C-kielestä. Lisäksi mukana on piirteitä Java-, Perl-, ja C++ -kielestä muutamine täysin omine PHP-lisäyksineen.

PHP:n avulla voidaan käyttää monia tietokantoja. Näistä mainittakoon PostgreSQL, MySQL ja Oracle. PHP on avoin ohjelmisto, ja sen lähdekoodi on vapaasti saatavilla ilmaiseksi.

Yleisimmin PHP:tä käytetään Unix/Linux-alustalle asennetun Apache Web -palvelimen kanssa. PHP:tä voi käyttää kuitenkin myös Windows-alustalla hyödyntäen vaikkapa Access-tietokantaa.

PHP-skriptit näyttävät tekijän kannalta tavallisilta HTML-dokumenteilta sillä erolla, että mukana on tietyillä tageilla erotettuna HTML-koodista komentoja. [8, s. 12 - 17]

3 KESKIKOSKEN LOMAMÖKKIEN INTERNET-SIVUJEN VAATIMUSMÄÄRITTELY

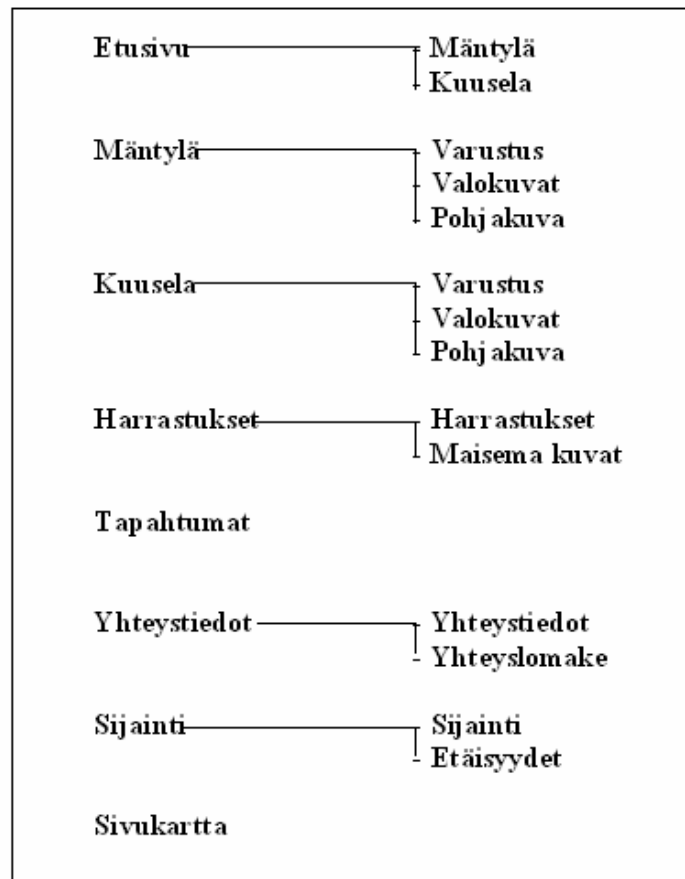
Tämä määrittely on tarkoitettu kattamaan Keskikosken Lomamökkien Internet-sivujen toteutuksen. Määrittely kattaa Keskikosken Internet-sivujen muotoilun ja päivityksen määrittelyt.

Työ on insinöörityö. Työn tarkoituksena on tehdä Keskikosken Lomamökkien Internet-sivut. Sivuston luonti tapahtuu määrittämällä ensin sivuston rakenne ja sivukartta, minkä jälkeen luodaan sivupohjan runkorakenne ja väritykset. Seuraavaksi määritellään sivulle tuleva tekstisisältö ja kuvatiedostot. Lopuksi määritellään tietokantaan ja yhteyslomakkeeseen liittyvät toiminnot.

3.1 Yleiskuvaus

Keskikosken Lomamökit on perheyrittys, joka toimii Sonkakoskella. Yrityksen toimiala on lomamökkien vuokraus. Sen omistuksessa on korkeatasoisia loma-asuntoja, jotka sijaitsevat upeassa jokimaisemassa luonnon helmassa. Loma-asunnot on suunniteltu vuokrattavaksi ympäri vuoden.

Internet-sivujen hierarkia määräytyy sivuston hakemistorakenteen perusteella. Alla on esitetty sivujen sivukartan avulla hakemistorakenne. Sivut tulevat toimimaan yrityksen toiminnan esittely-, yhteydenotto- ja mainossivuina. Sivusto on kokonaisuus HTML-sivuja ja skriptejä, jotka yhdessä tuottavat staattiset selattavat ja ajan tasalla olevat WWW-sivut hakemistorakenteen ja asetustiedostojen perusteella. Seuraavassa on esimerkki sivuston hakemistorakenteesta (kuva 2).



Kuva 2. Sivuston hakemistorakenne

Tuotteen käyttäjiä ovat Internet-käyttäjät ja Keskkikosken Lomamökit. Käyttäjät löytävät sivut yrityksen WWW-osoitteella ja hakukoneille annetuilla hakusanoilla. Yrittäjä toimii sivuston ylläpitäjänä ja päivittäjänä. Internet-sivujen toimintaympäristönä toimii SuomiDNS Webhotelli. Sivuston on tarkoitus pyöriä Webhotellin tarjoamilla ohjelmatuilla.

3.2 Ongelmat ja rajoitukset

Yritys on aloittanut toimintansa puoli vuotta sitten, ja tällä hetkellä yrityksellä ei ole Internet-sivuja. Tuotteen myynnin ja markkinoinnin kannalta yhteystiedoilla ja lomakkeella varustetut Internet-sivut olisivat oiva tapa tehdä yrityksen toiminta tunnetuksi asiakkaille ja luoda yhteyksiä. Yrityksestä tietoa haluaville Internet-sivut olisi myös helposti löydettävissä.

Sovelluksen turvallisuusvaatimukset ja yleiset rajoitteet määräytyvät pitkälti SuomiDNS Webhotellin tarjoamien palvelujen perusteella. Sivuston päivitys suoritetaan WS FTP -ohjelmalla yrittäjän kotikoneelta palvelimelle, joten sivuston tulee olla helposti siirrettävissä.

3.3 Toiminnalliset vaatimukset

Internet-sivujen tulee olla toimivia ja käytettäviä ainakin seuraavilla selaimilla:

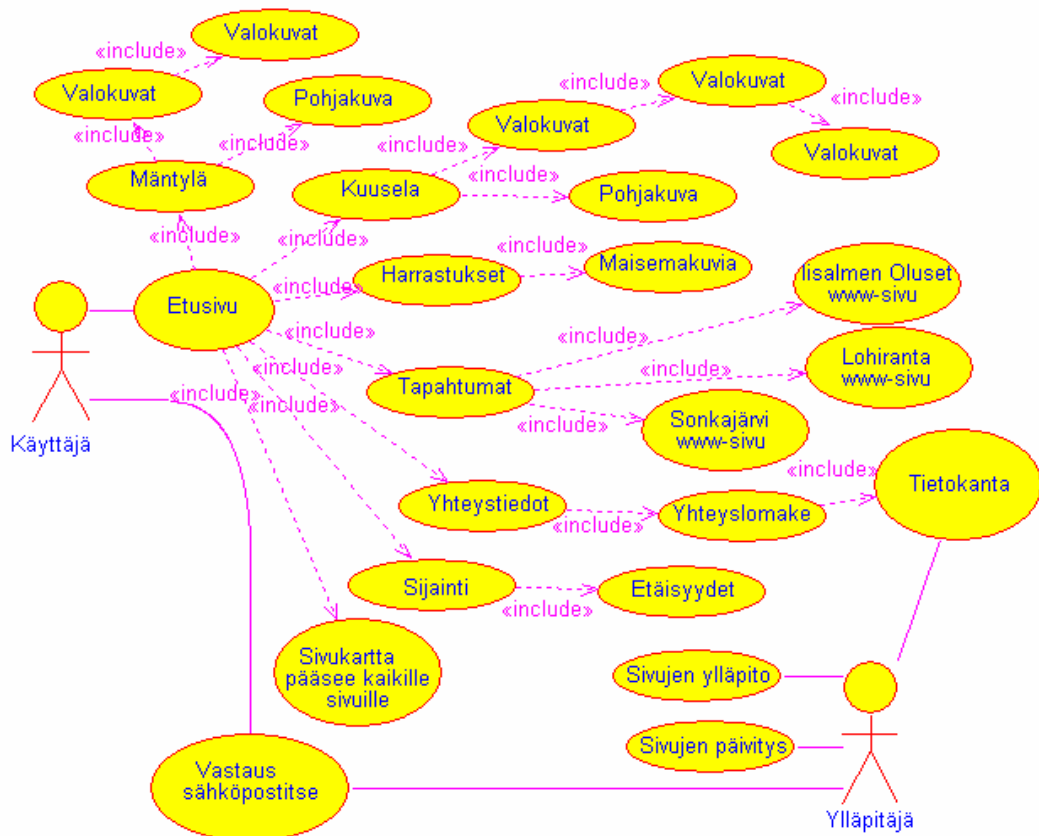
- Opera
- Internet Explorer.

Internet-sivujen vasemmassa laidassa olevassa päävalikossa on listattu sivuston pääsivujen otsikot linkkeinä hierarkkisessa järjestyksessä. Kullakin hetkellä auki oleva sivu näkyy valikossa tummennettuna. Alavalikot sijaitsevat pääsivujen sisältöosan ylälaidassa valikkopalkkina. Sivustossa liikutaan sivulta toiselle klikkaamalla valikosta halutun sivun otsikko. Tällöin näytetään otsikkoa vastaava sivu.

Tietokanta-palautelomake sijaitsee sivuston Yhteystiedot-alasivulla. Täyttämällä ja lähettämällä lomakkeen Internet-sivujen käyttäjä saa yrittäjältä vastauksen sähköpostitse lomakkeessa antamaansa sähköpostiosoitteeseen.

Käyttötapauskaavio

Internet-sivujen toiminnallisuus ja rakenne on kuvattu käyttötapauskaaviolla. Käyttötapauskaaviosta ilmenee, kuinka käyttäjä pystyy liikkumaan ja toimimaan sivuilla. Kaavio kuvaa myös sivuston ylläpitäjän tehtävät. Seuraavassa on esimerkki Internet-sivujen käyttötapauskaaviosta (kuva 3).



Kuva 3. Internet-sivujen käyttötapauskaavio

Käyttötapaukset

Toiminnot on jaettu kahdeksaan erilaiseen käyttötapaukseen. Käyttötapauksia ovat sivulle tuleminen, linkin valinta, alisivun linkin valinta, yhteyslomakkeen lähetys, tiedon nouto tietokannasta, sivujen ylläpito, sivujen päivitys ja vastaus sähköpostitse. Seuraavassa on esimerkit käyttötapauksista (taulukot 3, 4, 5, 6, 7, 8, 9 ja 10).

Taulukko 3. Käyttötapaus Sivulle tuleminen

KÄYTTÖTAPAUS:	Sivulle tuleminen
YHTEENVETO:	Käyttäjä tutustuu yrityksen Internet-sivujen etusivuun.
AKTORIT:	Sivuilla vierailleva käyttäjä.
EHDOT:	Sivujen valinta on oikein suoritettu.
KUVAUS:	Käyttäjälle avautuu yrityksen Internet-sivujen etusivu ja linkit.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa, Webhotelli ei toimi tai yrittäjä päivittää sivuja parasta aikaa.
LOPPUTULOS:	Käyttäjälle on avautunut etusivu.

Taulukko 4. Käyttötapaus Linkin valinta

KÄYTTÖTAPAUS:	Linkin valinta
YHTEENVETO:	Käyttäjä tutustuu Internet-sivujen alisivuun.
AKTORIT:	Sivuilla vierailleva käyttäjä.
EHDOT:	Linkki valittu.
KUVAUS:	Käyttäjälle avautuu valitsemansa alisivu ja linkit.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa, Webhotelli ei toimi tai yrittäjä päivittää sivuja parasta aikaa.
LOPPUTULOS:	Käyttäjälle on avautunut valittu alisivu.

Taulukko 5. Käyttötapaus Alasivun linkin valinta

KÄYTTÖTAPAUS:	Alasivun linkin valinta
YHTEENVETO:	Käyttäjä tutustuu Internet-sivujen alasivuun.
AKTORIT:	Sivuilla vieraileva käyttäjä.
EHDOT:	Linkki valittu.
KUVAUS:	Käyttäjälle avautuu valitsemansa alasivu ja linkit.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa, Webhotelli ei toimi tai yrittäjä päivittää sivuja parasta aikaa.
LOPPUTULOS:	Käyttäjälle avautuu valittu alasivu.

Taulukko 6. Käyttötapaus Yhteyslomakkeen lähetys

KÄYTTÖTAPAUS:	Yhteyslomakkeen lähetys
YHTEENVETO:	Käyttäjä lähettää yhteyslomakkeen.
AKTORIT:	Sivuilla vieraileva käyttäjä.
EHDOT:	Linkki valittu, lomake täytetty ja lähetetty.
KUVAUS:	Käyttäjälle avautuu valitsemansa alasivu, yhteyslomake ja linkit. Käyttäjä täyttää ja lähettää yhteyslomakkeen.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa, Webhotelli ei toimi tai yrittäjä päivittää sivuja parasta aikaa.
LOPPUTULOS:	Yhteyslomake lähetetty.

Taulukko 7. Käyttötapaus Tiedon nouto tietokannasta

KÄYTTÖTAPAUS:	Tiedon nouto tietokannasta
YHTEENVETO:	Yrittäjä hakee tiedot SuomiDNS:n cpaneelin MySQL-tietokanta taulusta.
AKTORIT:	Yrittäjä.
EHDOT:	Yrittäjä on kirjautunut cpaneeliin ja avannut MySQL-tietokanta taulun.
KUVAUS:	Yrittäjälle avautuu cpanelin ja tietokanta taulu.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa, Webhotelli ei toimi tai yrittäjä antaa väärän tunnuksen kirjautuessa.
LOPPUTULOS:	Yrittäjä saa tietokannasta tiedot lähetetyistä yhteyslomakkeista.

Taulukko 8. Käyttötapaus Sivujen ylläpito

KÄYTTÖTAPAUS:	Sivujen ylläpito
YHTEENVETO:	Yrittäjällä on Webhotellin palvelimella Internet-sivut.
AKTORIT:	Yrittäjä.
EHDOT:	Yrittäjä on ostanut SuomiDNS:ltä Webhotelli palvelun.
KUVAUS:	Yrittäjä sijoittaa Internet sivunsa Webhotellin palvelimelle.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa, Webhotelli ei toimi.
LOPPUTULOS:	Yrittäjällä on Internet-sivut.

Taulukko 9. Käyttötapaus Sivujen päivitys

KÄYTTÖTAPAUS:	Sivujen päivitys
YHTEENVETO:	Yrittäjä pitää Internet-sivunsa ajan tasalla päivittämällä.
AKTORIT:	Yrittäjä.
EHDOT:	Yrittäjä päivittää Internet-sivut Webhotellin palvelimelle WS FTB -ohjelman avulla.
KUVAUS:	Yrittäjä avaa WS FTB -ohjelman, kirjautuu ja siirtää päivitettävät sivut Webhotellin palvelimelle.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa, Webhotelli ei toimi tai yrittäjä kirjautuu väärin.
LOPPUTULOS:	Internet-sivut päivitetty.

Taulukko 10. Käyttötapaus Vastaus sähköpostitse

KÄYTTÖTAPAUS:	Vastaus sähköpostitse
YHTEENVETO:	Yrittäjä vastaa sähköpostitse yhteyslomakkeen lähettäneelle käyttäjälle.
AKTORIT:	Yrittäjä.
EHDOT:	Yrittäjä on lukenut yhteyslomakkeen tiedot tietokanta ja yhteyslomakkeen lähettäjä on antanut oikean sähköpostiosoitteen.
KUVAUS:	Yrittäjä lähettää sähköpostitse vastauksen käyttäjälle yhteyslomakkeessa lähetettyjen tietojen perusteella.
POIKKEUKSET:	Internetyhteydet eivät ole kunnossa tai sähköposti ei toimi.
LOPPUTULOS:	Käyttäjä saa vastauksen.

3.5 Ei-toiminnalliset vaatimukset

Keskeisinä yleisperiaatteina WWW-sivuston laadinnassa tulee pitää saatavuutta, löydettävyyttä ja palvelevuutta. Sivujen palvelevuutta voidaan arvioida sivujen sisällöllisen kattavuuden, palvelun tason, tiedon ajantasaisuuden ja informatiivisuuden kannalta. Löydettävyyden kannalta on oleellista, että yrityksen kotisivun (aloitussivun) ja Internet-sivujen tulisi olla koko ajan käytettävissä ja ajantasalla. Löydettävyyden kannalta on oleellista, että yrityksen kotisivun (aloitussivun) ja palvelun osoite ovat loogisia. Päivitykset eivät saa rikkoa sivustoilla olevaa HTML-tekstiä tai valikkorakennetta.

Tietoturvallisuuden kannalta on olennaista, että sivustojen sisältöä ja rakennetta pääsee muuttamaan ja päivittämään vain ylläpitäjä. Vanhennettuja eli päivitettyjä sivuja ei tallenneta. Muuten mitään erityisiä suojauksia ei ole.

Ylläpidon normaaleihin tehtäviin kuuluu lähinnä uusien sivujen/hakemistojen lisääminen sivustoon ja vanhojen muokkaaminen. Huollettavuus on otettava huomioon sivuja suunniteltaessa ja sivujen sisältö tulee olla helposti muokattavissa. Keskikosken Lomamökit huolehtii ylläpidosta.

Internet-sivuja tulee pystyä laajentamaan tarpeen mukaan lisäämällä uusia sivuja ja linkkejä sivustoon. Jos Internet-sivuja muunnellaan käyttöönoton jälkeen, insinööriyöntekijä ei vastaa järjestelmässä esiintyneistä virheistä.

3.6 Muut vaatimukset

Internet-sivujen suorituskyky määräytyy pitkälti käytössä olevan palvelimen kapasiteettirajoitusten perusteella. Sivuja suunniteltaessa on huomioitava, että ne latautuvat kohtuullisessa ajassa ja toimivat Webhotellin tarjoamilla ohjelmatuilla.

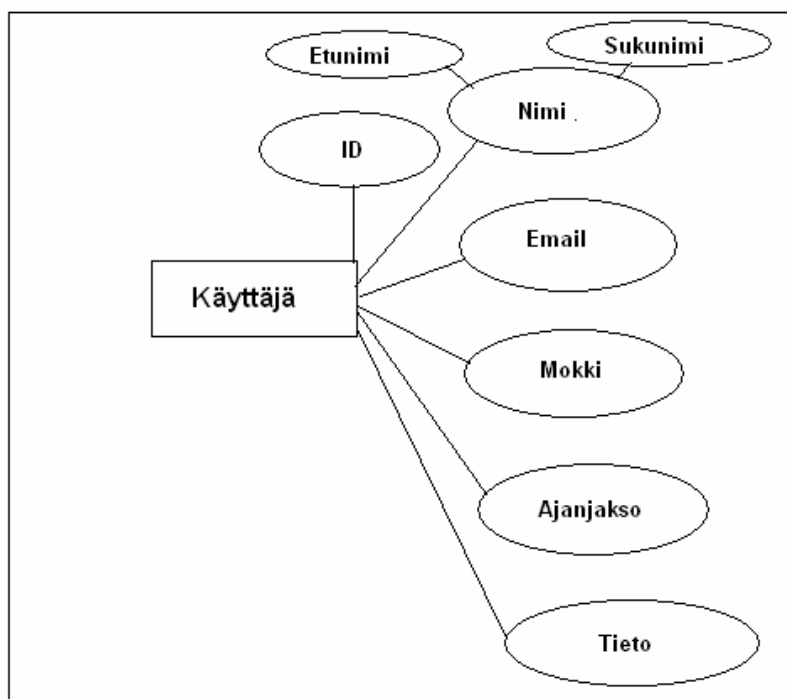
Rajapinnat määräytyvät Webhotelli SuomiDNS:n tarjoamien palveluiden mukaan. Webhotellin Cpaneelissa on käytettävissä Server API CGI, PHP Version 4.3.1 ja MySQL Version 4.0.12.

3.7 Käyttöliittymät

Internet-sivuja käytetään Internetin kautta selaimen välityksellä. Sivuja pääsee käyttämään menemällä sivujen etusivulle. Etusivulta on mahdollista siirtyä linkkien avulla alasivuille. Selaimen on tuettava ja sallittava ohjelmiston normaali toiminta.

3.8 Tiedot ja tietokanta

Tietokantana toimii SuomiDNS:n palvelimen tarjoama MySQL-tietokanta. Tietokanta ohjelmoidaan käyttöön SuomiDNS:n cpanelin kautta. Suunniteltu tietokanta on pienimuotoinen. Se sisältää yhden taulun, johon tallentuu yhteyslomakkeelta lähetetyt tiedot. Seuraavassa on esimerkki kuvaus ER-mallista (kuva 4).



Kuva 4. Kuvaus ER-mallista

Talletettavat tiedot

Tietokantaan talletettavien kuuden tietokentän tiedoista pakollisia ovat nimi, email ja mokki. Ajanjakso- ja tietokenttien arvo ei ole pakollinen. ID-kentän tietotyyppi tulee olla luku ja muiden kenttien teksti. Seuraavassa on esimerkki tietokantataulusta (taulukko 11).

Taulukko 11. Tietokantataulu

Taulu	
id	luku 11
nimi	teksti
email	teksti
mokki	teksti
ajanjakso	teksti
tieto	teksti

4 KESKIKOSKEN LOMAMÖKKIEN INTERNET-SIVUJEN SUUNNITTELU

Suunnittelun lähtökohtana olivat määrittelyssä päätetyt toiminnot ja ominaisuudet. Internet-sivujen ympäristönä toimii yrittäjän valitsema Webhotelli SuomiDNS, joka mahdollistaa sivujen toteutuksen kolmikerrosmallilla.

Pohjakerroksena toimivat tietokantapalvelin ja tietokanta. SuomiDNS-Webhotellissa oli käytettävissä MySQL-tietokanta, joka on käyttötarkoitukseen sopiva ja monipuolinen. MySQL tarjoaa rajapinnan PHP-ohjelmointikielille, joten se täyttää määrittelyssä päätetyt vaatimukset.

Seuraavaksi oli vuorossa välikerros, jossa on WWW-sivujen toiminnot. Webhotellin tarjosi käyttöön PHP-palvelinsivutekniikan, joka on suunniteltu dynaamisten WWW-dokumenttien luomiseen. PHP:ssä on sisäänrakennettu ohjelmointirajapinta MySQL-tietokannalle, ja se soveltuu yhtä hyvin yksinkertaisten ohjelmien kuin monimutkaisten tietojärjestelmien toteutustyökaluksi.

Lopuksi suunnittelimme asiakaskerroksen. WWW-sivujen ulkoasu toteutettaisiin PHP-, CSS-, JavaScript- ja HTML-tyyliä avulla. Jokaisen sivun runkorakenne olisi sama ja sisältökohta muunneltavissa. Hahmottelimme sivun runkorakenteen ja valitsimme sivuille tulevien linkkien, tekstin ja alueiden värikytykset. Sovimme, että yrittäjä toimittaa tarvittavat valokuvat ja valmiit tekstit sivuja varten. Lisäksi päätettiin laittaa etusivulle sisältöosaan tekstilaatikko, johon yrittäjä voisi sijoittaa haluamansa liikkuvan tekstin.

Työn toteutustavaksi sovittiin itsenäinen työskentely. Internet-sivut päivitetään viikoittain Webhotelli SuomiDNS:n palvelimelle, josta yrittäjä voi seurata työn etenemistä.

5 KESKIKOSKEN LOMAMÖKKIEN INTERNET-SIVUJEN TOTEUTUS

Valmiina materiaalina Internet-sivujen toteutusta aloitettaessa olivat yrittäjän ottamat ja valitsemat sähköisessä muodossa olevat valokuvat mökeistä ja niiden ympäristöstä. Tämä helpotti ja nopeutti sivuston ulkoasun ja rakenteen toteuttamista.

5.1 Tekninen ratkaisu

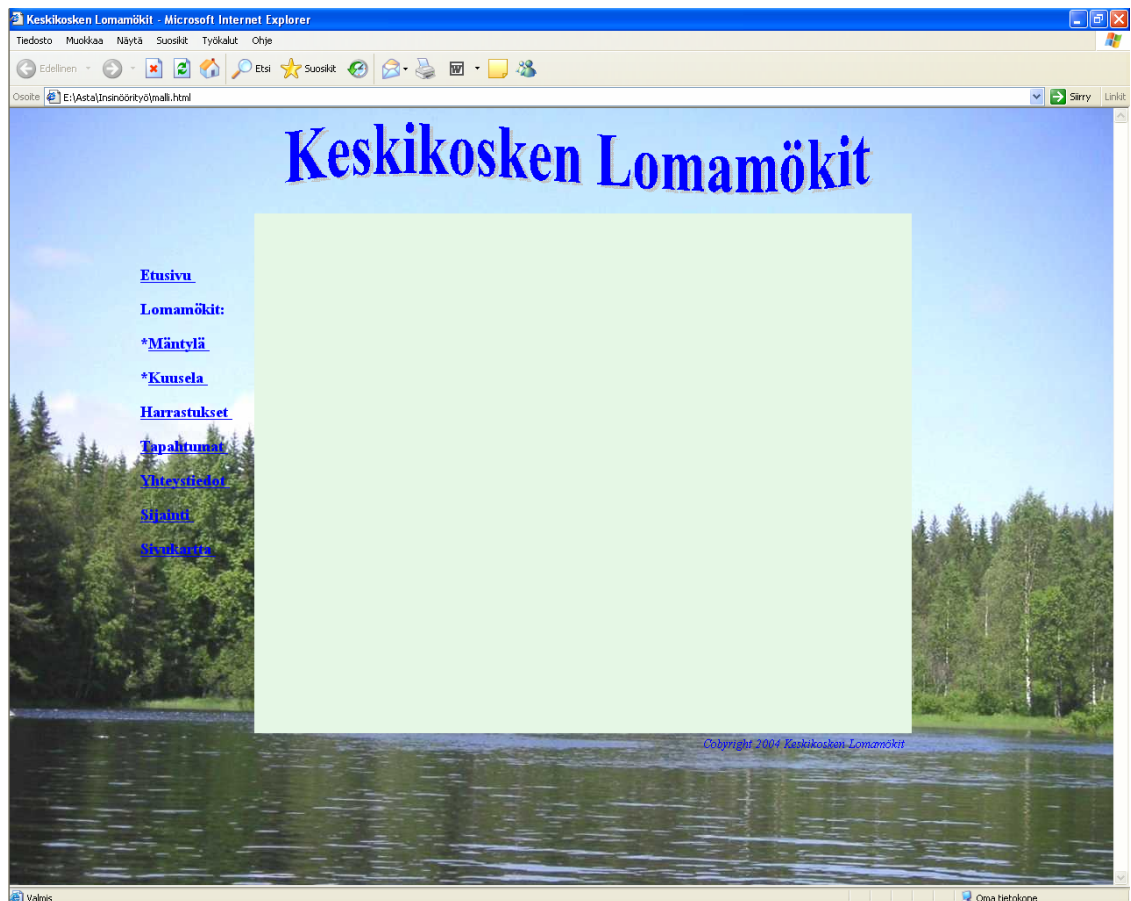
Kolmikerrosmallin mukaisessa ratkaisussa pohjakerroksena toimii MySQL-tietokantapalvelimessa (kehitysympäristössä versio 4.0.12) oleva tietokanta. Välikerros muodostuu WWW-palvelimesta ja sen PHP-laajennuksesta (kehitysympäristössä PHP 4.3.1) sekä palvelimella ajettavista WWW-dokumentteihin upotetuista PHP-skripteistä. Asiakaskerroksen muodostavat järjestelmän käyttäjän oma selain, CSS-tyylimäärittely ja selaimessa suoritettavat skriptit.

5.2 Sivupohjan toteutus

Sivupohja rakennettiin vastaamaan yrittäjän toivomaa ulkoasua ja rakennetta. Sivun taustakuvaksi laitettiin yrittäjän valitsema suvantomaisema-kuvatiedosto. Yläosaan tuli yrityksen nimilogo kuvatiedostona. Keskiosa sivusta on jaettu kahteen osaan. Vasemmanpuoleiseen osaan luotiin sivuston linkkivalikko ja oikeanpuoleiseen osaan sivun muuttuva sisältö. Tietokantalomake sijoitettiin Yhteyslomake-alasivulle. Tieto lomakkeesta siirretään palvelimella sijaitsevaan tietokantaan linkki.php ja tallennus.php avulla.

Sivuston tekstien, nimilogon ja linkkien väriksi päädyttiin laittamaan sininen. Muuttuvan sisältöosan taustaväriksi valittiin vaaleanvihreä, jonka värikoodi on #e5f7e5. Sivuston sivujen muuttuva sisältökohta luotiin jokaiselle sivulle erikseen valikkojen, linkkien, taulukoiden, kuvatiedostojen ja tekstitysten avulla.

Sijainti-sivulle tuli Suomen kartta, jossa on ilmaistu mökkien sijainti ja etäisyydet tietyistä kohteista. Tekijänoikeuslaki kieltää käyttämästä toisen tekemää valmista karttaa, joten kartta täytyi piirtää itse ja tallentaa kuvatiedostomuotoon. Mökeistä ei myöskään löytynyt sivuille sopivaa pohjakuvamallia, joten myös ne täytyi suunnitella, piirtää ja tallentaa kuvatiedostoksi. Seuraavassa on esimerkki sivunpohjan ulkoasusta (kuva 5).



Kuva 5. Sivunpohjan ulkoasu

Sivunpohjan koodi

Html-elementti on dokumentin tyyppi. Se aloittaa ja lopettaa dokumentin varsinaisen koodin. Seuraavassa on esimerkki html-elementin aloituksesta dokumentissa (taulukko 12).

Taulukko 12. Html-elementti

```
<html lang="fi">
```

Head-elementti on dokumentin tunniste. Se sisältää informaatiota, joka ei varsinaisesti näy sivuilla. Head-elementtejä ovat *title* ja *meta*. Title-tagin väliin tuli sivun nimi, joka näkyy selaimen yläreunassa statuspalkissa. Meta-elementissä on ilmoitettu ohjelma, jolla sivu on tehty, kuvaus sivusta ja sivuston hakusanoja. Seuraavassa on esimerkki head-elementin sisällöstä dokumentissa (taulukko 13).

Taulukko 13. Head-elementti

```
<head>  
<title>Sivun nimi</title>  
<meta http-equiv="Content-Type" content="text/html;  
charset=ISO-8859-1">  
<meta name="description" content="kuvaus">  
<meta name="keywords" content="mökki, mökit, matkailu, vuokramökit,  
yrityksen nimi">
```

Style-elementillä liitettiin tyylimäärittely dokumenttiin. Tässä dokumentissa on käytetty CSS-tyyliä. Span#otsake-kohdassa määriteltiin yrityksen nimilogon paikka ja koko yläosiossa. Span#menu-kohdassa määriteltiin linkkivalikon sijainti ja koko keskiosiossa. Span#sisalto-kohdassa määriteltiin dokumenttiin tulevan muuttuvan sisällön sijainti ja koko keskiosiossa. Spann#copyleft-kohdassa ilmoitettiin tekijänoikeustietojen paikka. Span#alalaita-kohdassa määriteltiin dokumentin loppumiskohta. Seuraavassa on esimerkki style-elementin sisällöstä (taulukko 14).

Taulukko 14. Style-elementti

```

<style type="text/css">
<!--
body { background: #e5f7e5 url('images/kuva.jpg') repeat-y;
a, a:visited          { color: blue;}
a:active              { color: red;}
a:hover               { }
a:hover.menu         { background-color: #e5f7e5; width: 100%;}
a:hover.sisalto      { background-color: #e5f7e5; width: 100%;}
span#otsake          { position: absolute; top: 15px; left: -315px
;width: 100}
span#menu             { position: absolute; text-align:left; padding:
60px; padding-left: 10px; padding-right: 10px;
top: 120; left: -490px; width: 70px; height: 150px;
border: groove 0px grey;}
span#keskustointi    { position: absolute; top: 0; left:50%; padding: 0;
margin: 0;}
span#sisalto          { position: absolute; left: -350px; top: 120px;
background-color: #e5f7e5; width: 750px;
span#copyleft        { position: absolute; right: 5px; bottom: -20px;
font-size: 90%; font-family: Times New Roman;}
span#alalaita        { position: relative; height: 10px;}
//-->
</style></head>

```

Body-elementti on dokumentin runko. Se sisältää informaatiota, joka näkyy sivuilla, eli otsikot, tekstit, kuvat ja muotoilu. Body-elementin attribuutteja ovat *background* (taustakuva), *bgcolor* (taustaväri), *visited* (käydyn linkin väri), *active* (aktivoidun linkin väri), *text* (tekstin väri) ja *hover* (linkin taustaväri). Body-elementin background-arvo ei periydy muille elementeille.

Dokumentin eri osioiden sisältö määriteltiin *span*-tagien sisälle. Tässä työssä kiinteitä osioita ovat `` ja ``. Otsake-osio sisältää logokuvatiedoston. Menu-osio sisältää sivuston linkkivalikon. Työn muuttuva sisältö tuli `` kohdan sisälle. Seuraavassa on esimerkki *body*-elementin sisällöstä (taulukko 15).

Taulukko 15. Body-elementti.

```
<body>
<span id="keskustointi">
<span id="otsake">

</span>
<span id="menu">
<h3><p><a href="index.html" class="menu">Etusivu&nbsp;</a></p>
<p><a target="showframe">Lomamökit:</a><dl compact>
<a href="sivu1.html" class="menu"><li>Mäntylä&nbsp;</li></p>
<p><a href="sivu2.html" class="menu"><li>Kuusela&nbsp;</li></p></dl></p>
<p><dl compact>
<a href="sivu3.html" class="menu">Harrastukset&nbsp;</a></p>
<p><a href="sivu4.html" class="menu">Tapahtumat&nbsp;</a></dl></p>
<p><a href="sivu5.html" class="menu">Yhteystiedot&nbsp;</a></p>
<p><a href="sivu6.html" class="menu">Sijainti&nbsp;</a></p>
<p><a href="sivu7.html" class="menu">Sivukartta&nbsp;</a></p></h3>
</span>
<span id="sisalto">Muuttuva sisältö</span>
<span id="copyleft">
<i><font color="blue">Copyright by Tekijä</font> </i>
</span></body></html>
```

5.3 Lomakkeen toteutus

Lomakkeen ulkoasun luonnissa noudatettiin yksinkertaista ja selkeää tyyliä. Lomakkeeseen laitettiin, jo aiemmin määrittelyssä päätetyt tietokantaan tallentuvat kentät. Nimi- ja sähköpostitiedoille laitettiin täytettävät tekstikentät, joihin mahtuu 56 merkkiä. Mökkien valintamahdollisuus toteutettiin valintanapeilla. Ajanjakso-kohtaan laadittiin valintalista, jossa on valittavana 52 viikkoa. Lopuksi laitettiin vielä tekstialue, johon käyttäjä voi kirjoittaa haluamansa viestin yrittäjälle. Seuraavassa on esimerkki yhteyslomakkeesta (kuva 6).

Täytä tähän yhteystietosi:

Nimi:

Sähköpostiosoite:

Valitse mökki josta haluat tietoa:

Mäntylä Kuusela

Valitse haluamasi ajanjakso:

▼

Haluamasi tiedot:

Kuva 6. Yhteyslomake

Lomakkeen koodi

Dokumenttiin tehtiin lomake *form*-koodilla. Lomakkeen aloittavassa *form*-koodissa määritellään, millä tavalla lomakkeeseen annetut tulokset käsitellään. Lomakkeen kentät eli täytettävät aukko- ja tekstikentät määriteltiin *input*-koodilla. Kentän tyyppi määriteltiin *input type* -määritelmällä. Tässä lomakkeessa on käytetty kentän arvoja *input text* (tekstikenttä), *input radio* (valintanappi), *input select* (valintalista) ja *input textarea* (tekstilaatikko). Kentät määriteltiin koodilla *input name* (nimi), *input value* (oletusarvo) ja *input size* (koko). Seuraavassa on esimerkki *form*-koodista (taulukko 16).

Taulukko 16. *Form*-koodi.

```
<form>
<form action="tallennus.php" method="post">
Nimi: <input type="text" name="nimi" value="nimi" size="56">
Email:<input type="text" name="email" value="email" size="56">
Valitse haluamasi mökki:
<label for=mantyla>&nbsp;&nbsp;&nbsp;<input type="radio" name="mokki"
value="mantyla">Mäntylä</label>
<label for=kuusela>&nbsp;&nbsp;&nbsp;<input type="radio" name="mokki"
value="kuusela">Kuusela</label>
Valitse ajanjakso:
<select size="1" name="ajanjakso">
<option selected>-----Valitse -----</option>
<option value>Ajanjakso 1</option>
...
<option value>Ajanjakso 52</option>
</select>
Haluamasi tieto:
<textarea type="text" name="tieto" cols="36" rows="5"></textarea>
```

Painike `input type="submit"` lähettää lomakkeen tulokset action-kohtassa määritellylle ohjelmalle ja `input type="reset"` tyhjentää kentät oletusasetuksiin. Painike `input type="hidden"` on piilokenttä. Piilokentällä lähetetään tietoa lomakkeen mukana lomaketta käsittelevälle ohjelmalle ilman, että käyttäjä näkee tietoa. Seuraavassa on esimerkki painikkeiden koodista (taulukko 17).

Taulukko 17. Painikkeiden koodi

```
<input type="submit" name="submit" value="Lähetä">
<input type="reset" name="reset" value="Tyhjennä">
<input type="hidden" name="redirect" value="index.html">
</form>
```

5.4 Tietokannan toteutus

Seuraavassa on esimerkki Webshotelli SuomiDNS:n Cpanelin tietokantauuluun tulevien tietojen sisältö (kuva 7).

Sarake	Tyyppi	Attribuutit	Tyhjä	Oletusarvo	Lisätiedot	Toiminto					
id	int(11)		Ei		auto_increment						
nimi	text		Ei								
email	text		Ei								
mokki	text		Ei								
ajanjakso	text		Kyllä	NULL							
tieto	text		Kyllä	NULL							

Kuva 7. Cpanelin tietokantataulu

Webshotellin palvelimen Cpaneliin luotiin MySQL-tietokanta ja taulu. Luontivaiheessa määritettiin taulun kentät ja tietotyypit. Perusavain-kentäksi laitettiin id-kenttä. ID-, nimi-, email- ja mokki-kentille asetettiin ominaisuusarvo *not null* (arvo tarvitaan). Taulutyypiksi valittiin MYISAM. Seuraavassa on esimerkki tietokantataulun MySQL-koodista (taulukko 18).

Taulukko 18. MySQL-tietokantataulun koodi.

```
CREATE TABLE `taulu` (  
  `id` INT(11) NOT NULL AUTO_INCREMENT ,  
  `nimi` TEXT NOT NULL ,  
  `email` TEXT NOT NULL ,  
  `mokki` TEXT NOT NULL ,  
  `ajanjakso` TEXT ,  
  `tieto` TEXT,  
  PRIMARY KEY ( `id` )  
 ) TYPE = MYISAM ;
```

5.5 Yhteyden luominen tietokantaan

Linkki.php on PHP-sovellus, joka määrittää polun webpalvelimella sijaitsevaan tietokantaan ja suorittaa yhteyden avaamisen palvelimella sijaitsevaan MySQL -tietokantaan.

Ensimmäiseksi linkki.php-koodissa määritetään SQL-palvelimen osoite, käyttäjä, salasana ja tietokannan nimi. Seuraavaksi luodaan yhteys palvelimeen käyttäjätunnuksen ja salasanan avulla. Yhteyden luonnin onnistuessa yhdistetään tietokantaan. Sovellus antaa virheilmoituksen, jos yhteyttä ei saada palvelimeen tai tietokantaan. Seuraavassa on esimerkki Linkki.php-koodista (taulukko 19).

Taulukko 19. Linkki.php-koodi

```
<!doctype html public "-//W3C//DTD HTML 4.0 //EN">
<html>
<head>
<title>linkki</title>
</head>
<body>
<?
$db_host = "lochalhost";
$db_user = "käyttäjä";
$db_pass = "salasana";
$db_name = "tietokannan nimi";

$db = @mysql_connect($db_host,$db_user,$db_pass);
if (!$db)
{
print "<p><b>Yhteyttä ei saatu.</p></b>";
exit;
}
@mysql_select_db ($db_name) or die ("<p><b>Yhteyttä ei saatu.</p></b>")
?>
</body>
</html>
```

5.6 Tietokantaan tallentaminen

Tallennus.php on PHP-sovellus, joka tallennetaan samaan hakemistoon itse lomakkeen kanssa. Ensiksi tutkimme, mitä lomakkeenkäsittelijäsovellusta käytetään, eli tuleeko käyttäjä sallitulta palvelimelta. Kontrollointi on tärkeää, ettei sovellusta voida käyttää väärin. Sovelluksen ajaminen päättyy, jos palvelinta ei tunnisteta "www.sallitun_domainin_nimi.fi" nimiseksi. Muussa tapauksessa lomakkeenkäsittelijän suoritus jatkuu siirtyen keräämään ja tallettamaan käyttäjän syötteet muuttujien arvoiksi.

Seuraavaksi tarkastetaan kenttien nimi, email ja mökki syötteet. Tyhjä kenttä palauttaa käyttäjälle virheilmoituksen ja linkin, jonka avulla hän pääsee jatkamaan lomakkeen täyttämistä. Lomakesovelluksen ajaminen pysäytetään *exit-funktion* avulla. Seuraavassa on esimerkki valintaruutujen tarkastus (taulukko 20).

Taulukko 20. Valintaruutujen tarkastus

```
<?
if (empty($_POST['nimi']))
{
echo "<p><h3>Nimi tietosi jäi täyttämättä.</h3></p>";
echo "\n<p>Palaa takaisin <a href=\"javascript:history.back(1)\">
<b>lomakkeeseen</b> </a>\n täyttääksesi puuttuva tieto.</body>\n</html>";
exit;
}
if (empty($_POST['email']))
{
echo "<p><h3>Email tietosi jäi täyttämättä.</h3><p>";
echo "\n<p>Palaa takaisin <a href=\"javascript:history.back(1)\">
<b>lomakkeeseen</b> </a> täyttääksesi puuttuva tieto.\n</body>\n</html>";
exit;
}
```

```

if (empty($_POST['mokka']))
{
echo "<p><h3>Mökki, josta haluat tietoa jäi valitsematta.</h3><p>";
echo "\n<p>Palaa takaisin <a href=\"javascript:history.back(1)\">
<b>lomakkeeseen</b> </a> täyttääksesi puuttuva tieto.\n</body>\n</html>";
exit;
}

```

Yhteyden saamiseksi tietokantaan ajetaan linkki.php ja muodostetaan yhteys, ellei yhteyttä saada tulostuu virheilmoitus. Seuraavassa on esimerkki yhteyden muodostuksesta (taulukko 21).

Taulukko 21. Yhteyden muodostus

```

include "linkki.php";

$db = mysql_connect($db_host,$db_user,$db_pass);

mysql_select_db ($db_name) or die ("Yhteyttä ei saatu.");
if (!$db)
{
print "<p class=\"error\">Yhteyttä ei saatu.</p>";
exit;
}

```

Suoritetaan turvallisuussyistä tavallisille tekstikentille tarkastus, jotta mitään ei-toivottuja tageja tai SQL-lauseita ei pääsisi tekstikenttien läpi tietokantaan. Tagien riisunta suoritetaan *strip_tagson-funktiolla*. Seuraavaksi ajetaan tekstisyötteet *mysql_espase_string-funktion* kautta, joka korvaa tekstissä mahdollisesti olevat lainausmerkit lisäämällä niiden eteen kenoviivan (slash).

Tämäkin funktio parantaa turvallisuutta vaikeuttaen SQL-hyökkäyksen tekemistä (SQL injections). Ilman sitä saamme virheilmoituksen tietokantapalvelimelta, eikä datan tallennus onnistu, koska lainausmerkki aiheuttaa syntaksivirheen kyselyä suoritettaessa. Seuraavassa on esimerkki turvallisuustarkastus koodista (taulukko 22).

Taulukko 22. Turvallisuustarkastus

```
$_POST['nimi'] = strip_tags($_POST['nimi']);
$_POST['email'] = strip_tags($_POST['email']);
$_POST['mokki'] = strip_tags($_POST['mokki']);
$_POST['ajanjakso'] = strip_tags($_POST['ajanjakso']);
$_POST['tieto'] = strip_tags($_POST['tieto']);

$_POST['nimi'] = mysql_escape_string($_POST['nimi']);
$_POST['email'] = mysql_escape_string($_POST['email']);
$_POST['mokki'] = mysql_escape_string($_POST['mokki']);
$_POST['ajanjakso'] = mysql_escape_string($_POST['ajanjakso']);
$_POST['tieto'] = mysql_escape_string($_POST['tieto']);
```

Suoritetaan lisäkysely, ja tallennetaan lomakkeelta peräisin oleva tarkastettu data tietokantatauluun nimeltä taulu. Seuraavassa on esimerkki lisäkyselyn suorittaminen ja tallennus tietokantaan (taulukko 23).

Taulukko 23. Lisäyskyselyn suorittaminen ja tallennus tietokantaan

```
$query = "INSERT INTO taulu (id, nimi, email, mokki, ajanjakso, tieto)
VALUES ('','.$_POST['nimi'].','.$_POST['email'].','.$_POST['mokki'].',
'.$_POST['ajanjakso'].','.$_POST['tieto'].')";

$result = mysql_query($query);
```

Rekisteröinnin onnistuessa tulostaa kiitostekstin ja linkkin etusivulle sekä sulkee tietokannan. Rekisteröinnin epäonnistuu tulostaa virheilmoituksen ja sulkee tietokannan. Seuraavassa on esimerkki tallennuksesta ja tietokantayhteyden sulkemisesta (taulukko 24).

Taulukko 24. Tallennus ja tietokantayhteyden sulkeminen

```
if (!$result)
{
die("<p>Rekisteröinti epäonnistui".mysql_error());
}
else
{
echo "<p>Kiitos yhteydenotostasi!</p>";
echo "\n<p>\nPalaa tästä takaisin<a href=\"index.html\"><b>etusivulle</b></a>";
mysql_close($db); }
?>
```

5.7 Ajastetun tekstikentän toteutus

Etusivun sisältökohtaan sijoitetaan tekstikenttä, johon tulee ajastettu teksti. Tekstikenttä luodaan JavaScriptin avulla. Teksti kirjoittuu tekstikenttään kirjoituskonetyylillä. Seuraavassa on esimerkki JavaScriptin aloitusmäärittelystä (taulukko 25).

Taulukko 25. JavaScriptin aloitusmäärittely

```
<script language ="JavaScript">
<!-- begin
var max=0;
```


Luodaan *function textlist()* -kohdassa tekstilista ja määritellään sen sisältö. Määrittelyn sisälle kirjoitetaan teksti, minkä halutaan näkyvän tekstikentässä. *unction textticker()* -kohdassa määritetään ajoitus, millä teksti kirjoittuu tekstikenttään. Seuraavassa on esimerkki funktioiden sisällöstä (taulukko 26).

Taulukko 26. Funktioiden sisältö

```
function textlist()
{
max=textlist.arguments.length;
for (i=0; i<max; i++) this[i]=textlist.arguments[i];
}
tl=new textlist
(
" Hinnat alkaen 67 €/ vrk/ 2 hlö, lisähenkilöt a-hinta 10 €/ vrk !  "
);
var x=0; pos=0;
var l=tl[0].length;
unction textticker()
{
document.tickform.tickfield.value=tl[x].substring(0,pos)+"_";
if(pos++==l)
{
pos=0;
setTimeout("textticker()",1000);
x++; if(x==max) x=0;
l=tl[x].length;
} else
setTimeout("textticker()",150);
}
// end --></script>
```

Tehdään vielä tekstikenttä, johon teksti tulostuu etusivun WWW-dokumentissa. Määritellään tekstikentän nimeksi tickform, tyypiksi teksti ja kooksi 60 merkkiä. Seuraavassa on esimerkki tekstikentän luonti (taulukko 27).

Taulukko 27. Tekstikentän luonti

```
<script language="JavaScript">
<!--
document.write("<form name=\"tickform\">");
document.write("<input type=\"text\" name=\"tickfield\" size=\"60\">");
document.write("</form>");
textticker();
//--></script>
```

5.8 Sivujen testaus

WWW-dokumenttien laadintavaiheessa sivujen toimintaa testattiin insinööriyöntekijän kotikoneella Internet Explorerilla. PHP-koodilla laadittuja palvelinsivuja ei voitu vielä tässä vaiheessa testata, koska ei ollut tarpeellisia ohjelmia.

Yrittäjä otti WWW-sivuosoitteen SuomiDNS:n Webhotelliin. Tässä vaiheessa pääsimme testaamaan WWW-sivut Internet-ympäristössä. Siirsimme WS FTP -ohjelmalla palvelimelle ensiksi testausta varten vain WWW-dokumentit, ja ne toimivat moitteettomasti.

MySQL-tietokannan valmistuttua siirsimme WS FTP -ohjelmalla PHP:llä laaditut palvelinsivut linkki.php ja tallennus.php palvelimelle. Järjestelmän toimintaan saaminen vaati aikapaljon työtä, sillä yhteyttä ei aluksi saatu tietokantaan. Tämä johtui tiedon puutteesta ja vääristä palvelin-, käyttäjä-, salasana- ja tietokanta-määritelmistä linkki.php:ssä.

Internet-sivujen ja tietokannan valmistuttua kokonaisuudessaan siirrettiin sivut yrittäjän kotikoneelle. Sieltä ne ladattiin WS FTP -ohjelmalla SuomiDNS:n palvelimelle. Tässä vaiheessa tehtiin myös sivujen virallinen testaus. Samalla tutkittiin, toimiiko tietojen tallennus MySQL-tietokannassa sijaitsevaan tietokantatauluun oikein. Sivusto ja tietokantayhteys toimivat moitteettomasti.

5.9 Sivujen ylläpito ja käyttöönotto

Virallisesti Keskikosken Lomamökkien Internet-sivut otettiin käyttöön 21.1.2005 yrittäjän kotona siirtämällä lopullinen versio SuomiDNS:n palvelimelle. Käyttöönoton yhteydessä yrittäjälle luovutettiin myös CD-levylle poltettu alkuperäinen versio Internet-sivuista ja kirjallisessa muodossa olevan MySQL-tietokannan käyttöohje. Käyttöohjeessa on opastettu, kuinka tietokantataulusta SuomiDNS:n palvelimelta noudetaan sinne yhteyslomakkeelta lähetetyt kyselyt sekä miten tietokantataulua muunnellaan ja tyhjennetään.

Käyttöönoton yhteydessä sivujen ylläpitovastuu siirtyi insinööriyöntekijältä yrittäjälle. Insinööriyöntekijä toimitti yrittäjälle sivujen ylläpitoa varten CCS-, HTML- ja PHP-materiaalia. Materiaali sisältää ohjeet insinööriyöntekijän WWW-dokumenttien laadinnassa tekemiin koodeihin ja niiden muokkaamiseen. Insinööriyöntekijä opettaa yrittäjän muokkaamaan ja päivittämään WWW-sivuja antamansa materiaalin avulla.

6 YHTEENVETO

Tavoitteena oli luoda toimivat ja yrittäjän toiveita vastaavat Internet-sivut. Niiden tuli kuvastaa yrityksen toimintaa ja olla rakenteeltaan yksinkertaisen selkeät. Näihin tavoitteisiin päästiin hyvin. Keskikosken Internet-sivut ovat toiminnassa ja kattavat niille määritellyt vaatimukset.

Internet-sivujen toteutuksessa käytettiin HTML-, CSS-, JavaScript-, PHP- ja SQL-koodausta. Yhteyslomakkeelta tallentuvat tiedot palvelimella sijaitsevaan MySQL-tietokantaan. Yrittäjä pystyy kotikoneeltaan päivittämään Internet-sivut aina tarpeen vaatiessa ja katsomaan yhteyslomakkeelta lähetetyt kyselyt SuomiDNS:n palvelimen MySQL-tietokantataulusta.

Suurimmat ongelmat työn tekemisessä liittyivät ehdottomasti aikatauluihin. Olin viikolla päivät töissä ja illat koulussa, joten insinööriyölle jäi aikaa vain viikonloppuisin. Toinen ongelma oli tiedon löytäminen ja saanti. Vaikka PHP-ohjelmoinnista ja MySQL-tietokannasta on olemassa paljon materiaalia, niin ei kuitenkaan insinööriyössä tarvittu osalta. Vaikeinta työssä oli yhteyden saaminen tietokantaan ohjelmoinnin avulla. Minulla ei ollut aikaisempaa kokemusta MySQL-tietokannasta ja PHP-ohjelmoinnista, joten tiedon etsiminen, opiskelu ja soveltaminen otti aikansa. Työn edistymisen myötä kehittyi valtavasti myös projektityöskentely ja PHP-ohjelmointitaidot.

Yrittäjältä saadun palautteen mukaan Internet-sivut ovat vastanneet odotuksia ja toimineet hyvin. Sivujen päivittäminen ja muokkaaminen on myös helppoa.

LÄHDELUETTELO

- 1 Fowler, Martin - Kendal, Scott. UML. Jyväskylä, Docendo Finland Oy, 2002.
- 2 Koskimies, Kari. Oliokirja. Jyväskylä, Satku, 2000.
- 3 Lahtonen, Tommi. SQL. Jyväskylä. Docendo Finland Oy, 2003. ISBN 951-846-092-2.
- 4 Lahtonen, Tommi. SQL. Vantaa, Tummavuoren kirjapaino Oy, 2002.
- 5 Rantala, Jarmo. Hyvään verkkopalveluun. Tampere, Tammer-Paino Oy, 2002.
- 6 Heinisuo, Rami. PHP ja MySQL. Jyväskylä, Talentum Media Oy, 2001. ISBN 951-762-770-X.
- 7 Packalén, Jukka. Johdatus HTML -kieleen. 1997 [WWW -dokumentti] <<http://www.helsinki.fi/atk/www/ohjeet/wwwopas/html-opas.html>> (Luettu 15.9.2004)
- 8 Pääkkönen, Pekka. CSS - Tyylimäärittely. 1998 [WWW -dokumentti] <<http://www.hytti.uku.fi/~ppaakkon/css.html>> (Luettu 03.11.2004)
- 9 Rantala, Ari. PHP. Porvoo, Docendo Finland Oy, 2002. ISBN: 951-846-147-3