

SATAKUNNAN AMMATTIKORKEAKOULU

Teemu Jönkkäri

SEKVENSSIKUVAUSJÄRJESTELMÄN TIETOKANNAN JA  
JULKAISUJÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS

OHJELMISTOTEKNIikka

2007

# SEKVENSSIKUVAUSJÄRJESTELMÄN TIETOKANNAN JA JULKAISUJÄRJESTELMÄN SUUNNITTELU JA TOTEUTUS

Jönkkäri, Teemu  
Satakunnan ammattikorkeakoulu  
Tekniikka Pori  
Tietotekniikan koulutusohjelma  
Maaliskuu 2007  
Ekholm, Ari  
UDK: 004.6, 004.738.1  
Sivumäärä 43

Avainsanat: julkaisujärjestelmät, MySQL, PHP, relaatiotietokannat, valokuvaus

---

Opinnäytetyössä suunniteltiin relaatiotietokanta uudentyyppiselle sekvenssikuvausta hyödyntävälle havaintovälineelle. Opinnäytetyön toisena tavoitteena oli suunnitella ja toteuttaa sekvenssikuvausjärjestelmää hyödyntävä Internet-sivusto, jonka tärkeimpänä tehtävänä on toimia käyttöliittymänä tietokantaan kootun datan tutkimuksessa. Käyttöliittymän piti mahdollistaa kuvatallenteiden etsiminen suuresta datamäärästä ja niiden luokittelu, käyttäjän niistä löytämien ominaisuuksien perusteella. Internet-sivuston tuli toimia samalla myös Karhukamera-projektin tavoitteista ja etenemisestä kertovana verkkosivustona.

Opinnäytetyötä kehitettiin tiiviissä yhteistyössä Karhukamera-projektiryhmän kanssa, jolta saatu palaute otettiin suunnittelussa jo varhain huomioon. Erityisen suuri vaikutus projektiryhmän palautteella oli relaatiotietokannan suunnittelussa.

Opinnäytetyönä kehitetyn relaatiotietokannan ja Internet-sivuston toimivuus ja käytettävyys täyttivät hyvin projektin alussa asetetut tavoitteet ja tarjoavat hyvän perustan järjestelmän kehittämiseksi myös tulevaisuudessa.

# DESIGN AND IMPLEMENTATION OF A DATABASE AND WEB-BASED PUBLISHING SYSTEM FOR A TIME-LAPSE PHOTOGRAPHY SYSTEM

Teemu Jönkkäri  
Satakunta University of Applied Sciences  
School of Technology Pori  
Degree Programme in Information Technology  
March 2007  
Ekholm, Ari  
UDK: 004.6, 004.738.1  
Pages: 43

Key words: MySQL, PHP, relational databases, time-lapse photography, web-based publishing system

---

The aim of this thesis was to design a relational database for a new kind of observation aid that uses time-lapse photography. Another aim was to develop a website to serve as a user interface for the system. As a user interface the website's key feature is to let the user search visual recordings from the database and mark them with labels that best describe the visual property of the recording. The website must also serve as a conventional website for publishing news and general information about the project Karhukamera.

The thesis was designed and implemented in collaboration with a project group. The feedback from the project group was taken into account from the start. The greatest influence on the thesis was feedback about the features of the relational database.

The relational database and website have been proven to work properly and that they fulfill the requirement specifications. They also provide a solid base for the deployment of the system in the future.

# SISÄLLYS

1 JOHDANTO .....	8
1.1 Opinnäytetyön tausta.....	8
1.2 Opinnäytetyön tarkoitus ja tavoitteet .....	8
1.3 Teoreettinen viitekehys .....	9
2 MYSQL.....	9
2.1 MySQL 5.0.....	10
2.2 InnoDB .....	12
3 PHP .....	14
4 HTML/XHTML JA CSS .....	16
4.1 HTML/XHTML .....	16
4.2 CSS.....	17
5 JÄRJESTELMÄARKKITEHTUURI.....	18
6 TIETOKANNAN SUUNNITTELU JA TOTEUTUS .....	21
7 KÄYTTÖLIITTYMÄ .....	22
7.1 Käyttäjäautentikointi .....	22
7.2 Verkkosivuston sivulaatikat.....	25
7.3 Tallenteiden hakeminen tietokannasta .....	26
7.3.1 Hakulomake.....	27
7.3.2 Haku .....	28
7.3.3 Hakutulokset.....	29
7.4 Kuvatallenne ja siihen liittyvän informaation esittäminen.....	30
7.4.1 Kuvatallenne.....	30
7.4.2 Kuvaukset .....	31
7.4.3 EXIF-tiedot.....	32
7.4.4 Sää tiedot .....	33
7.4.5 Muut tiedot .....	35
8 VIDEOT KUVATALLENTEIDEN KÄYTÖN TUKENA .....	36
9 VARMUUSKOPIOINTI.....	37
10 PROJEKTIN NYKYTILA JA TULEVAISUUS.....	38
10.1 Projektin nykytila .....	38

10.2	Projektin tulevaisuus .....	39
10.3	Internet-sivuston kehittäminen tulevaisuudessa.....	39
10.4	Relaatiotietokannan kehittäminen tulevaisuudessa .....	40
11	YHTEENVETO .....	40
	LÄHTEET .....	41
	LIITTEET	

## TERMILUETTELO

AJAX	Asynchronous JavaScript and XML on vuorovaikutteisten verkkosovellusten kehittämiseen käytetty tekniikka.
EXIF	Exchangeable image file format.
FTP	File Transfer Protocol on käyttöjärjestelmäriippumaton tiedonsiirtomenetelmä, jolla voidaan siirtää dataa kahden tietokoneen välillä.
GPS	Global Positioning System on Yhdysvaltojen puolustusministeriön kehittämä ja rahoittama reaaliaikainen satelliittipaikannusjärjestelmä.
HTTP	Hypertext Transfer Protocol on tiedonsiirtoprotokolla, jota käytetään yleisimmin selainohjelman ja Internet-palvelimen väliseen tiedonsiirtoon.
IETF	Internet Engineering Task Force on Internet-protokollien standardoinnista vastaava organisaatio.
IMAP	Internet Message Access Protocol on sähköpostien lukemiseen tarkoitettu protokolla.
JavaScript	Internet-selaimessa suoritettava komentosarjakieli.
Komentosarjakieli	Ohjelmointikieli, jossa ohjelmakoodia ei käännetä erikseen ennen ohjelman suorittamista, vaan se ajetaan sellaisenaan ohjelman suoritusvaiheessa.
LDAP	Lightweight Directory Access Protocol on hakemistopalvelujen käyttöön tarkoitettu verkkoprotokolla.
METAR	Meteorological Aerodrome Report on säähavainnon standardoitu tallennus- ja viestitysmuoto, jota käytetään lentosääpalvelussa.
Perl	Practical Extraction and Report Language on Larry Wallin kehittämä alustariippumaton tulkettava proseduraalinen komentosarjakieli.
POP3	Post Office Protocol version 3 on yksinkertainen ja suosittu sähköpostiprotokolla.

Python	Tulkattava ohjelmointikieli, jolla on vahva dynaaminen tyyppitys ja automaattinen roskienkeruu. Python tukee oliopohjaista, proseduraalista ja funktionaalista ohjelmointitapaa.
RAID	Redundant Array of Independent Disks.
RDBMS	Relational Database Management System eli relaatiotietokannan hallintajärjestelmä.
SOAP	Simple Object Access Protocol on tietoliikenneprotokolla, jonka pääasiallisena tehtävänä on mahdollistaa proseduurien etäkutsu. Se pohjautuu XML-kieleen ja toimii useiden eri protokollien yli, joista tärkein on HTTP-protokolla.
SQL	Structured Query Language on IBM:n kehittämä standardoitu kyselykieli, jolla tietokantaan voi tehdä erilaisia hakuja, muutoksia ja lisäyksiä.
Web Service	Verkkopalvelulla tarkoitetaan yleisesti sovellusliittymää, johon voidaan ottaa yhteys verkon kautta ja käyttää etäjärjestelmän tarjoamia palveluita.
XHTML	Extensible Hypertext Markup Language on HTML:stä kehitetty Internet-sivujen merkintäkieli, joka täyttää XML:n muotovaatimukset.
XML	Extensible Markup Language on W3C:n spesifikaatioihin perustuva menetelmä rakenteisten dokumenttien kirjoittamiseen ja merkkaukseen. Se on samalla myös metakieli, jolla voidaan määritellä uusia merkkaukieliä.

# 1 JOHDANTO

## 1.1 Opinnäytetyön tausta

Sekvenssikuvauksen kehittäminen aloitettiin kesällä 2004, ensisijaisena tarkoituksena tarjota reaaliaikainen säätieto Ulvilan observatorion kaukoputken käyttäjille, Internetin välityksellä. Hyvin pian projektin aloittamisen jälkeen havaittiin sekvenssikuvauksen hyödyntämätön potentiaali tieteellisenä havaintovälineenä. (Viitapohja 2007a.)

Sekvenssikuvaukseen yhdistää korkean resoluution digitaalisen valokuvan, videon, tietokannan ja erilaiset mitattavat suureet kuten säätiedot havaintovälineeksi, joka on tehokas työväline pitkäkestoisten ilmiöiden havainnointiin, joissa visuaalisen havainnon merkitys on suuri. (Viitapohja 2007b.)

Sekvenssikuvauksen kehittämiseksi, yllämainitun määritelmän kaltaiseksi toimivaksi havaintovälineeksi, perustettiin Karhukamera-projekti. Projektin tavoitteena on kehittää järjestelmä, joka luo visuaalisen tallenteen havaittavasta kohteesta ja liittää siihen tiedot vallitsevista olosuhteista sekä havainnon kannalta tärkeistä suureista, jotka edesauttavat havaittavan ilmiön ymmärtämistä (Viitapohja 2007b).

Mahdollisuus osallistua sekvenssikuvaukseen perustuvan havaintovälineen kehittämiseen suunnittelemalla ja toteuttamalla järjestelmään tarvittava relaatiotietokanta sekä Internet-sivusto herättivät mielenkiintoni projektia kohtaan, sillä näin siinä erinomaisen mahdollisuuden syventää osaamistani niin tietokantojen suunnittelusta kuin Internet-sivustojen kehittämisestäkin.

## 1.2 Opinnäytetyön tarkoitus ja tavoitteet

Opinnäytetyön tavoitteena on suunnitella sekvenssikuvaukseen hyödyntävälle järjestelmälle relaatiotietokanta, jossa on järjestelmän tarvitsemat ominaisuudet. Tietokan-



taan tallennetaan kuvien ja videoiden lisäksi myös iso joukko erilaisia mitattavia suureita kuten aika-, paikka- ja säätietoja. Tietokannan tulee olla myös joustava ja helposti laajennettavissa, jotta kehitettyä järjestelmää voitaisiin tulevaisuudessa käyttää erilaisiin digitaalista valokuvausta hyödyntäviin projekteihin ja tutkimuksiin. Tietokannan pitää sisältää lisäksi järjestelmän käyttöliittymänä toimivan Internet-sivuston tarvitsemat taulut.

Opinnäytetyön toisena tavoitteena on suunnitella ja toteuttaa sekvenssikuvausjärjestelmälle Internet-sivusto, jonka tärkeimpänä tehtävänä on toimia käyttöliittymänä tietokantaan kootun datan tutkimuksessa. Käyttöliittymän pitää mahdollistaa kuvataallenteiden etsiminen suuresta datamäärästä ja niiden luokittelu, käyttäjän niistä löytämien ominaisuuksien perusteella. Käyttäjän pitää pystyä myös kommentoimaan tallenteita vapaamuotoisesti.

Internet-sivuston tarkoituksena on myös esitellä kehitettävää havaintovälinettä ja kertoa Karhukamera-projektin etenemisestä yhteistyökumppaneillemme sekä muille kehitettävästä järjestelmästä kiinnostuneille, suomeksi ja englanniksi.

### 1.3 Teoreettinen viitekehys

Opinnäytetyössä yhdistyvät relaatiotietokannan suunnittelu MySQL-tietokantaohjelmistolle, toimintalogiikan toteuttaminen PHP-komentosarjakielillä sekä Internet-sivuston ulkoasun suunnittelu ja toteutus.

## 2 MYSQL

MySQL-tietokantaohjelmisto on MySQL AB:n kehittämä relaatiotietokannan hallintajärjestelmä eli RDBMS.

MySQL AB on ruotsalainen, vuonna 1995 perustettu yritys, jonka nykyinen pääomistaja on Oracle Corporation. Yritystä pidetään yhtenä suurimmista ns. avoimen lähdekoodin yhtiöistä maailmassa (Wikipedia 2007a). MySQL AB:n kehittämästä MySQL-tietokantaohjelmistosta on nykyisin tarjolla kaupallinen MySQL Enterprise Server ja ilmainen MySQL Community Server.

MySQL-tietokantaohjelmisto on monipuolinen, joustava, turvallinen ja sen suorituskyky riittää suurtenkin tietokantojen hallintaan. MySQL noudattaa asiakas-palvelin-arkkitehtuuria, jossa sovellukset eivät pääse koskaan käsittelemään tietokantaa suoraan vaan käsittely tapahtuu aina palvelinohjelman kautta. MySQL-tietokantaohjelmisto on myös kevyt ylläpidettävä, eikä se vaadi päivittäistä huolenpitoa. (Heinäsuu 2003, 36.)

MySQL-tietokantaohjelmisto on erityisen suosittu Internet-ympäristössä juuri edullisuutensa, helppokäyttöisyytensä ja turvallisuutensa ansiosta. MySQL-tietokantaohjelmisto onkin nykyään selvästi suosituin ilmainen ja avoimen lähdekoodin tiedonhallintajärjestelmä.

MySQL-tietokantaohjelmiston tällä hetkellä suositeltava versio on loppuvuodesta 2005 julkaistu 5.0, jota käytetään myös sekvenssikuvauksjärjestelmässä.

## 2.1 MySQL 5.0

MySQL sai 5.0-versiossa useita, jo kaupallisista tietokantaohjelmistoista tuttuja ominaisuuksia, kuten triggerit, näkymät ja tallennettavat funktiot. Ohjelmiston yhteensopivuutta parannettiin SQL-kyselykielen nykyisen ANSI/ISO SQL:2003-standardin kanssa, uhraamatta kuitenkaan sen nopeutta ja luotettavuutta. MySQL-tietokantaohjelmistossa on myös suuri valikoima erilaisia käytettävyyttä parantavia laajennuksia SQL-kyselykieleen. (MySQL 5.0 Reference Manual 2005, 13-27.)

Näkymät ovat ilmentymiä, joihin voidaan koota informaatiota useammasta taulusta. Näkymien avulla voidaan yksinkertaistaa ja nopeuttaa tietojen hakemista tietokannasta. Näkymien avulla on myös mahdollista päivittää yksittäisten taulujen sisäl-

töä, mutta mikäli näkymään on liitetty informaatiota useammasta taulusta, näiden samanaikainen päivittäminen ei onnistu. Näkymät kuitenkin helpottavat tietokantojen suunnittelijoiden työtä. Tietokantaa käyttävän sovelluksen tekijälle voidaan esimerkiksi tarjota taulujen sijasta valmiita näkymiä käytettäväksi, jolloin sovellukselta voidaan rajata pääsy vain valittuihin taulun tietueisiin. (MySQL 5.0 Reference Manual 2005, 1123-1130.)

Triggeri eli liipaisin on nimetty tietokantaobjekti, joka liitetään tauluun. Kun taululle tapahtuu ennalta määritelty tapahtuma, triggeri aktivoituu ja se suorittaa sille ohjelmoidun tehtävän. Ennalta määrittelyksi tapahtumaksi voidaan valita esimerkiksi INSERT-, UPDATE- tai DELETE-lause. Lisäksi voidaan määrittellä halutaanko triggeri suorittaa ennen vai jälkeen määritellyn tapahtuman. (MySQL 5.0 Reference Manual 2005, 1116-1120.)

Tallennettavat funktiot ja proseduurit ovat tietokantaohjelmistoon valmiiksi ohjelmoituja tehtäviä ja toimintoja, joita voidaan käyttää esimerkiksi kyselyjen suorittamiseen tietokannassa. Tallennettavilla funktioilla ja proseduureilla voidaan suorittaa toimintoja hyvin yksinkertaisista tehtävistä aina monimutkaisiin ehtolauseisiin (Listaus 1). Tallennettavilla funktioilla ja proseduureilla voidaan tarjota esimerkiksi asiakassovellukselle valmiita yksinkertaistettuja kyselyjä, jotka muuten vaatisivat asiakassovellukselta useiden monimutkaisten kyselyjen suorittamista.

```
mysql> create function hei (  
    -> maailma varchar(25) )  
    -> returns varchar(25)  
    -> return concat('Hei ', maailma, '!');  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> select hei('maailma');  
+-----+  
| hei('maailma') |  
+-----+  
| Hei maailma!   |  
+-----+  
1 row in set (0.01 sec)
```

Listaus 1. Esimerkki tallennettavan funktion toteuttamisesta MySQL:ssä

Tallennettavilla funktiolla saavutetaan useita etuja kuten yhdenmukaisuutta ja turvallisuutta. Asiakassovellukselta voidaan estää suora pääsy tietokannan tauluihin ja tarjota vain valmiita funktioita käytettäväksi. Tämän kaltaiset turvallisuutta parantavat ominaisuudet korostuvat esimerkiksi pankkimaaailmassa. Tallennettavilla funktioilla voidaan myös parantaa asiakassovelluksen suorituskykyä kun osa laskennasta siirtyy tietokantapalvelimelle ja dataliikenteen määrä tietokantaohjelmiston ja asiakassovelluksen välillä vähenee. (MySQL 5.0 Reference Manual 2005, 984.)

MySQL 5.0 sisältää useita ns. tietokantamoottoreita, joista ehdottomasti tärkeimmät ovat MyISAM ja InnoDB. MySQL:n oletustietokantamoottorina on MyISAM, joka on erinomainen valinta kun järjestelmältä vaaditaan pelkästään kirjoitus- ja lukunopeutta. MyISAM-tietokantamoottori ei kuitenkaan tue kehittyneempiä ominaisuuksia kuten transaktioita ja viiteavaimia vaan tällöin pitää käyttää kehittyneempää InnoDB-tietokantamoottoria. MySQL-tietokantaohjelmistossa voi vapaasti käyttää eri tietokantamoottoreita käyttäviä tauluja samassa kyselylauseessa. (MySQL 5.0 Reference Manual 2005, 873-881.)

## 2.2 InnoDB

InnoDB-tietokantamoottorin on kehittänyt suomalaisen Innobase Oy. InnoDB-tietokantamoottori on lisensoitu samalla GPL-lisenssillä kuin MySQL Community Server.

InnoDB-tietokantamoottori on suunniteltu raskaisiin tuotantojärjestelmiin ja prosessoimaan tehokkaasti suuria datamääriä. Sen CPU-tehokkuudelle ei löytyne vastinetta muista levymuistia käyttävistä relaatiotietokantamoottoreista. (MySQL 5.0 Reference Manual 2005, 840.)

InnoDB-tietokantamoottori ylläpitää omaa puskuriallasta välimuistissa olevalle datalle ja indekseille. Toisin kuin MyISAM, joka tallentaa kaikki taulut omiin tiedostoihinsa, InnoDB tallentaa taulujen tiedot tauluavaruuteen, joka voi koostua useista tiedostoista tai jopa alustamattomista levyosioista, jolloin tiedoston enimmäiskoko ei

riipu käyttöjärjestelmän asettamista rajoitteista. (MySQL 5.0 Reference Manual 2005, 880.)

InnoDB on transaktioturvallinen relaatiotietokantamoottori, joka pystyy toipumaan virhetilanteista. InnoDB-tietokantamoottorissa on myös ominaisuuksia, jotka parantavat monen käyttäjän järjestelmien toimintojen yhtäaikaaisuutta ja suorituskykyä. Siitä on esimerkiksi lukkiutumaton SELECT-lauseen toteutus sekä tietuekohtainen lukitusmahdollisuus. (MySQL 5.0 Reference Manual 2005, 880-881.)

InnoDB-tietokantamoottori noudattaa tiukasti ns. ACID-sääntöjä, joiden noudattaminen takaa tietokannan eheyden kaikissa tilanteissa. Monet järjestelmät joustavat ACID-säännöistä tavoitellessaan parempaa suorituskykyä. ACID on akronyymin englannin kielen sanoista, jotka merkitsevät atomisuutta, eheyttä, eristyneisyyttä ja pysyvyyttä. Atomisuudella tarkoitetaan tässä yhteydessä sitä, että transaktio pitää pystyä joko suorittamaan tai vaihtoehtoisesti peruuttamaan kokonaan. Eheydellä tarkoitetaan, että tietokanta tulee siirtää aina yhdestä eheästä tilasta toiseen. Eristyneisyys tarkoittaa, että transaktiot eivät saa vaikuttaa järjestelmässä toisiinsa vaan niiden tulee toimia kuin ne olisivat järjestelmässä yksin. Lisäksi suoritettujen transaktioiden pitää säilyä järjestelmässä kaikissa mahdollisissa virhetilanteissa. (Gray 1981, 1-6; Wikipedia 2007b.)

InnoDB-tietokantamoottori tukee myös viiteavainrajoitteita ja estää kaikki INSERT- ja UPDATE-operaatiot, jotka pyrkivät luomaan vierasavaimen lapsitauluun, jos sillä ei ole vastinetta isätaulussa. Viiteavainrajoitteet määritellään lapsitauluun käyttäen FOREIGN KEY -lauseen ON UPDATE- ja ON DELETE -alilauseita. (MySQL 5.0 Reference Manual 2005, 897-898.)

Kun tietokantaan luodaan uusi InnoDB-tietokantamoottoria käyttävä taulu, pitää CREATE TABLE -lauseeseen lisätä asetusparametri ENGINE=InnoDB (MySQL 5.0 Reference Manual 2005, 894.)

### 3 PHP

PHP on komentosarjakieli, jonka kehitys alkoi vuonna 1994. Rasmus Lerdorf julkaisi ensimmäisen virallisen version PHP-komentosarjakielestä vuonna 1995. Hän yhdisti siinä joukon Perl-komentosarjakielisiä skriptejä ja C-ohjelmointikielellä kirjoitettuja CGI-ohjelmia.

PHP-komentosarjakieli perustuu avoimen lähdekoodin periaatteille ja kielen ympärille kehittyikin nopeasti yhteisö, joka on kehittänyt PHP-komentosarjakielestä nykyaikaisen ja täysiverisen tulkattavan ohjelmointikielen.

PHP 3. versio, joka kehitettiin ensimmäisten alkeellisten versioiden jälkeen, alkoi jo muistuttaa ominaisuuksiltaan kielen nykyisiä versioita. Siinä oli kehittynyt jäsentelijä, joka mahdollisti erittäin hyvän laajennettavuuden ja tuen useille tietokantaohjelmistoille, protokollille ja sovellusliittymille. Sen suorituskyky ei kuitenkaan riittänyt palvelemaan uusia kehittyneitä palveluita. (PHP Documentation Group 2007a.)

Zeev Suraski ja Andi Gutmans alkoivat kehittää PHP:n 3. version julkaisun jälkeen PHP-komentosarjakielelle uutta suorituskykyisempää ja modulaarisempaa ydintä. Kehitystyö onnistui hyvin ja vuonna 2000 julkaistun PHP 4. version ytimenä oli uusi ja tehokas Zend Engine. PHP:n 4. versio tarjosi lisäksi uusina ominaisuuksina kielen mm. istunnot, tulostuksen puskuroinnin ja edistyneemmän syötteiden hallinnan. PHP:n 4. versio lisäsi myös tuen monille uusille palvelinohjelmistoille. (PHP Documentation Group 2007a.)

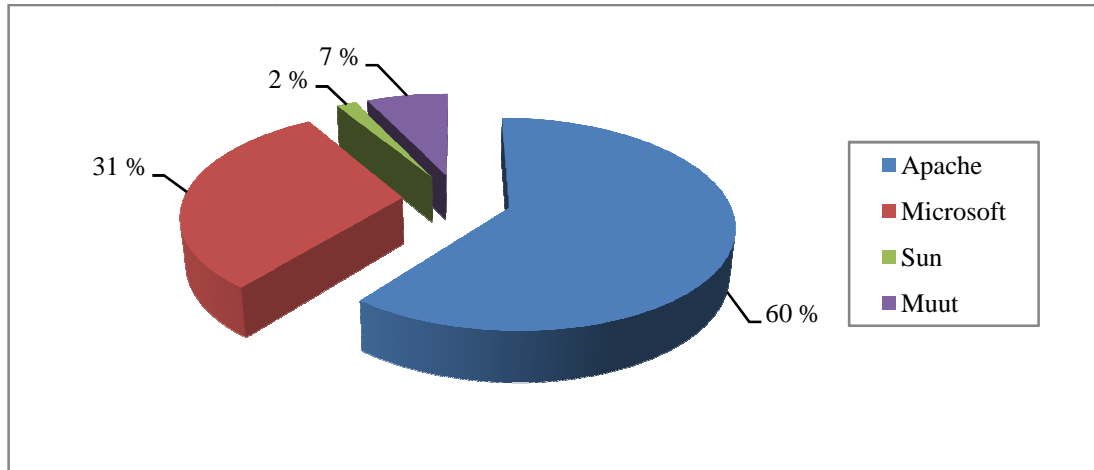
PHP on periaatteessa tukenut olio-ohjelmointia jo 3. versiosta asti, mutta sen tuki on ollut hyvin rajoittunutta ja puutteellista. Vasta vuonna 2005 julkaistu PHP:n 5. versio toi kieleen kehittyneen tuen olio-ohjelmoinnille.

PHP:n 5. versio, jonka ytimenä on Zend Engine II, mahdollistaa C++- ja Java-ohjelmointikielten tavoin standardin tavan luoda ja käsitellä olioita. PHP:n uudistuneet olio-ohjelmointiominaisuudet muistuttavat voimakkaasti Java-ohjelmointikielen vas-

taavia toteutuksia. Uusia ominaisuuksia ovat mm. hajotin ja muodostin metodien standardointi, abstraktien luokkien ja metodien lisääminen kieleen sekä luokkamuuttujien ja metodien suojaaminen. Standardin poikkeuskäsittelyn lisääminen kieleen on myös yksi tärkeimmistä uusista ominaisuuksista, koska se mahdollistaa aiempaa vakaampien ja tehokkaampien ohjelmien kehittämisen. (Suraski 2004.)

Muita merkittäviä uudistuksia PHP 5. versiossa on mm. parantunut tuki XML-merkintäkielisten dokumenttien käsittelylle sekä Web Service -palveluille kuten SOAP. PHP:n uusin virallinen versio on marraskuussa 2006 julkaistu 5.2.0.

PHP on nykyisin hyvin pitkälle alustariippumaton ja sen voi asentaa niin Linux-, Mac-, Unix- kuin Windows-ympäristöönkin. Se toimii myös yleisimpien Internet-palvelinohjelmistojen kuten Apachen ja Microsoft ISS:n kanssa, jotka hallitsevat yhdessä yli 90 prosentin osuutta Internetissä toimivista palvelimista (Kuvio 1). Yleisimmin PHP on asennettuna Apache-palvelimen sisään moduulina, mutta PHP:n suorittaminen CGI-prosessina on myös mahdollista.



Kuvio 1. Ohjelmistovalmistajien osuudet Internetissä toimivista palvelimista joulukuussa 2006 (Netcraft 2006).

PHP-komentosarjakielen ehdottomana vahvuutena on sen laaja tuki kolmansien osapuolten ohjelmointirajapinnoille. PHP:ssa on mm. monipuolinen tuki erilaisille tietokantaohjelmistoille, joiden käytölle löytyy omia luokkakirjastoja. Omilla luokkakirjastoilla tuettuja tietokantaohjelmistoja ovat mm. MySQL, PostgreSQL, SQLite, Sybase ja Oracle. PHP:ssa on lisäksi käytettävissä ODBC-luokkakirjasto, joka tar-

joaa yhtenäisen rajapinnan useille muille suosituille tietokantaohjelmistoille kuten IBM DB2, Solid ja MS Access. Suorituskyvyssä ODBC-luokkakirjasto ei kuitenkaan pärjää ohjelmistokohtaisille luokkakirjastoille. PHP:ssa on myös hyvä tuki yleisimmille viestintäprotokollille kuten HTTP, FTP, POP3 ja IMAP.

PHP:n kuten muidenkin ns. upotettavien komentosarjakielten ensisijaisena tavoitteena on tarjota helppo ja nopea kehitysympäristö ohjelmistokehittäjille. PHP-komentosarjakieltä voidaan kuitenkin käyttää myös ohjelmoimaan rutiininomaisia prosesseja esimerkiksi Linux-ympäristössä, samaan tapaan kuin vaikkapa Pythonilla. Tämä mahdollisuus yksinkertaistaa ja selkeyttää kokonaisuuden hallintaa, kun kaikki toiminnallisuudet voidaan toteuttaa yhdellä komentosarjakiielellä.

## 4 HTML/XHTML JA CSS

Nykyaikaisessa Internet-sivustotuotannossa erotetaan dokumentin fyysinen rakenne ja muotoilu. Dokumentin rakenne toteutetaan HTML- tai XHTML-merkkauksielellä ja muotoilu lisätään CSS-tyylisivulla. World Wide Web Consortium (W3C) kehittää, ylläpitää ja julkaisee näiden keskeisten Internet-julkaisutekniikoiden standardeja.

### 4.1 HTML/XHTML

Tim Berners-Lee ja Robert Caillau hahmottelivat ensimmäisen version HTML-merkkauksielestä vuonna 1989 pyrkiessään kehittämään Euroopan ydintutkimuskeskukselle (CERN) uutta, yhtenäistä formaattia dokumentoinnille. Vuoteen 1996 asti HTML-merkkauksielen versioiden standardoimisesta vastasi IETF, mutta tämän jälkeen standardointivastuu on ollut WC3-konsortiolla. Viimeisin standardoitu versio HTML-merkkauksielestä on HTML 4.01. (Boumphrey, Greer & Raggett 2000, 8.)



HTML-merkkäuskieli oli erinomainen väline yksinkertaisten dokumenttien esittämiseen, mutta Internetin käytön nopea laajeneminen ja uudet päätelaitteet kuten puhelimet ovat osoittaneet HTML-merkkäuskielen ongelmat. HTML-merkkäuskielen mahdollistama huono ja epätarkka kuvaaminen vaatii dokumenttia tulkitsevalta laitteelta monimutkaista ja raskasta laskentaa, joka ei ole mahdollista esimerkiksi kevyissä kannettavissa päätelaitteissa. (Boumphrey ym. 2000, 26-27.)

Ongelman ratkaisuksi kehitettiin W3C-konsortion johdolla uusi XHTML-merkkäuskieli, joka yhdistää HTML- ja XML-merkkäuskielten parhaat ominaisuudet. XHTML käyttää HTML:n sanastoa ja XML:n kielioppia. W3C-konsortio julkaisi ensimmäinen XHTML 1.0 -standardin vuonna 2000. Vuonna 2001 julkaistiin nykyinen XHTML 1.1 -standardi. W3C-konsortio on jo pitkään suunnitellut uutta XHTML 2.0 -standardia joka tulee etääntymään HTML-merkkäuskielestä eikä se tule myöskään olemaan yhteensopiva aiempien XHTML-merkkäuskielen versioiden kanssa. Uuden XHTML 2.0 -standardin tarpeen taustalla on jatkuvasti monipuolistuva päätelaitteiden määrä sekä uuden ns. Semanttisen Webin kehittyminen. (Axelsson, Birbeck & Dubinko 2006, 11.)

XHTML-merkkäuskieli on suunniteltu yksinomaan rakenteen kuvaamiseen, joten dokumentin ulkoasun ja tyylin toteuttamiseen käytetään erillistä tyylimäärittelyä. Tähän tarkoitukseen on kehitetty kaskadinen tyyliohjelmäärittely, josta käytetään nimitystä CSS eli Cascade Style Sheets.

## 4.2 CSS

W3C-konsortio julkaisi ensimmäisen kaskadisia tyyliohjelmäärittelyjä koskevan CSS 1 -standardin vuonna 1996 ja sitä täydentävän CSS 2 -standardin vuonna 1997. Kaskadisuus tarkoittaa tässä yhteydessä sitä, että tyylimäärittelyjä voidaan periyttää ja korvata uusilla määrittelyillä. Dokumentissa olevan elementin näkyvyyttä ja tyyliä voidaan myös muotoilla monessa erillisessä tyyლისivussa. Käyttäjä voi myös korvata verkkosivulla määritellyn tyylin omalla tyylimäärittelyllä päätelaitteessaan.

CSS:n avulla erilaisille päätelaitteille voidaan määrittää sopiva tyyli joustavasti ja helposti, puuttumatta itse dokumentin rakenteeseen. Tämä parantaa huomattavasti tuotetun aineiston käytettävyyttä ja saavutettavuutta.

Kaskadisilla tyyliohjelmäärittelyillä voidaan kuvata, paitsi nähtävää, myös kuultavaa esitystapaa. Äänisyntetisaattoreita varten on määriteltävissä muun muassa äänen korkeutta, painotusta ja äänenväriä sääteleviä ominaisuuksia.

Web Standards Project -ryhmittymän suunnittelema Acid2-testi, joka on suunniteltu testaamaan verkkoselainten tukea HTML ja CSS standardeille, osoittaa miten puutteellisesti olemassa olevia standardeja nykyisin vielä tuetaan. Testin täysin läpäisseitä verkkoselaimia ovat ainoastaan Safari, Konqueror, Shiira, Opera ja iCab. Verkkoselainten huonon ja epätasaisen tuen vuoksi verkkosivuja ei voida kehittää pelkästään standardien mukaisiksi, vaan niitä joudutaan testaamaan useilla eri selaimilla ja tekemään selainkohtaisia korjausmäärittelyjä. (Wikipedia 2007d.)

## 5 JÄRJESTELMÄARKKITEHTUURI

Sekvenssikuvauksjärjestelmässä yhdistyy mm. sääpalvelu, kuvausyksikkö ja tallenteiden käsittely yhdeksi kokonaisuudeksi. Opinnäytetyöni keskittyy sekvenssikuvauksjärjestelmässä tuotettujen tallenteiden ja niihin liittyvän informaation hallintaan ja julkaisuun.

Järjestelmän sääpalveluna toimii tällä hetkellä ilmainen Wunderground-palvelu, mutta järjestelmä ei ota kantaa siihen missä ja milloin säätieta on tuotettu, joten säätieta-palvelua voidaan helposti vaihtaa. Parhaimpaan luotettavuuteen ja reaaliaikaisuuteen päästäisiin luonnollisesti omalla kameran läheisyyteen sijoitettavalla sääasemalla.

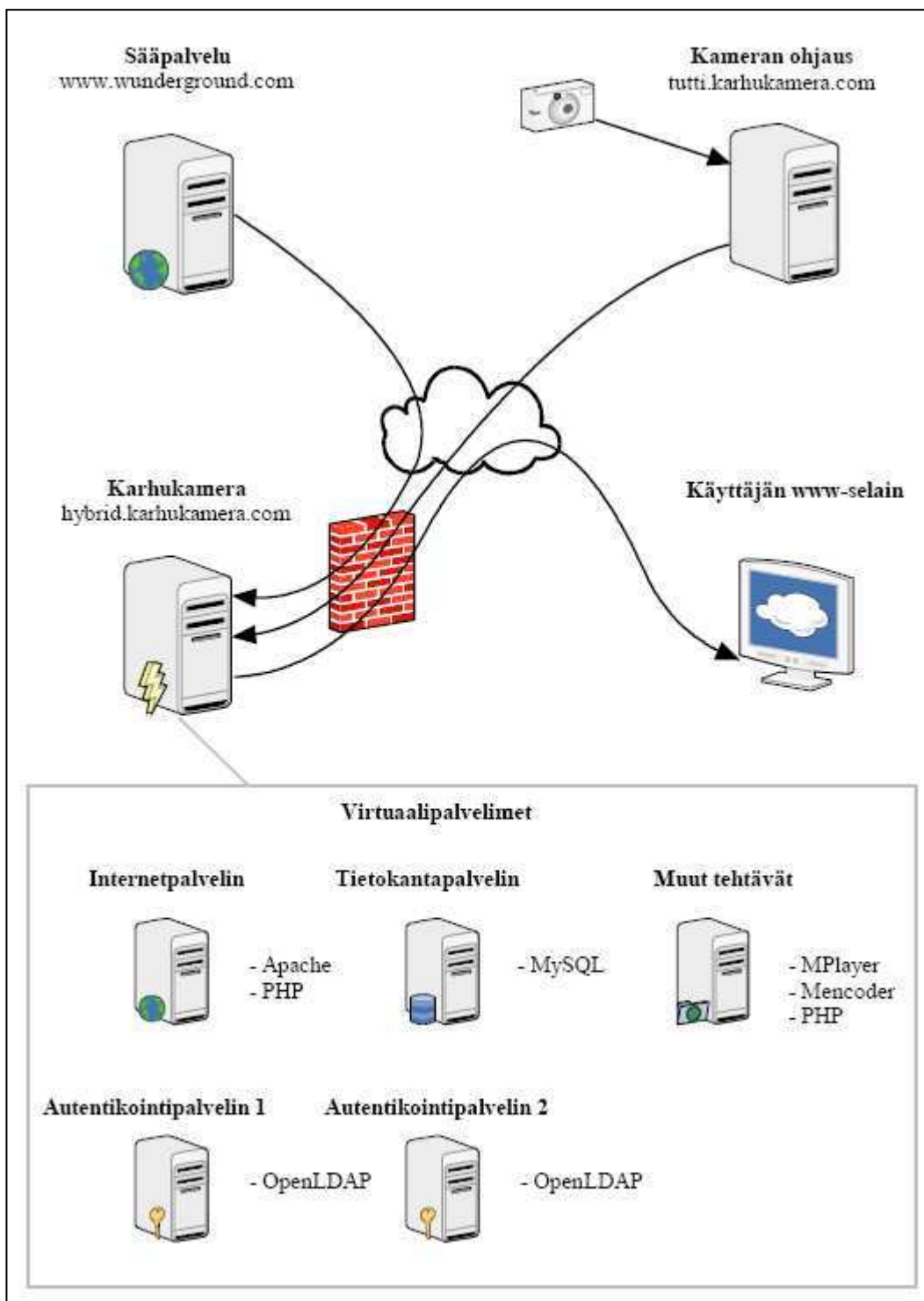
Sekvenssikuvauksjärjestelmä ei ota myöskään kantaa siihen miten, milloin tai missä järjestelmään tuotava kuvatalenne on luotu. Näin eri kameroiden käyttö on helppoa

ja järjestelmään voidaan tuoda tallenteita, jotka on kuvattu järjestelmän ulkopuolella. Kehitysvaiheessa järjestelmää on testattu ja testataan edelleen Ulvilan observatorion katolle sijoitetulla digikameralla. Digikameraa ohjataan tietokoneelle asennetulla Cam2Com-ohjelmistolla, joka on suunniteltu juuri digikameroiden etäohjaukseen. Digikameran ottamat kuvat tallennetaan ensin kameraa ohjaavalle tietokoneelle, josta ne siirretään ftp-yhteyden yli varsinaiselle Karhukamera-projektin palvelimelle. Järjestelmään on lisäksi tuotu kuvia kahdesta muusta kamerasta, jolloin on voitu todeta järjestelmän toimivan hyvin myös useammalla kameralla.

Karhukamera-projektin palvelinjärjestelmä koostuu yhdestä fyysisestä palvelimesta, jolla ajetaan useita erityistehtäviin keskittyviä virtuaalipalvelimia eli ns. loogisia palvelimia (Kuvio 2).

Virtuaalipalvelinjärjestelmä on toteutettu Xen Hypervisor -virtuaalikonemoottorin päälle. Xen on paravirtualisoiva x86-arkkitehtuurin virtuaalikonemoottori. Paravirtualisoinnilla tarkoitetaan laitteistovirtualisointia, jossa ei toteuteta prosessorin käskykantaan täydellisesti vaan virtualisoitavan käyttöjärjestelmän rakennetta muutetaan toimimaan virtuaalikonemoottorin käskykannalla.

Palvelinjärjestelmän palvelinten käyttöjärjestelmänä on Gentoo Linux, joka eroaa useista muista Linux-jakelupaketeista siinä, että se käännetään asennusvaiheessa suoraan lähdekoodista, jolloin käyttöjärjestelmä optimoituu aina juuri käytettävälle alustalle sopivaksi. Opinnäytetyöni kannalta tärkeimpiä ohjelmistoja palvelinjärjestelmässä ovat Apache 2.0 -palvelinohjelmisto, MySQL Community Server 5.0 -tietokantaohjelmisto ja PHP-komentosarjakieli.



Kuvio 2. Järjestelmäarkkitehtuuri.

## 6 TIETOKANNAN SUUNNITTELU JA TOTEUTUS

Sekvenssikuvauksjärjestelmän tietokantaohjelmistoksi valittiin Internet-ympäristössä suosittu, tehokkaaksi ja turvalliseksi tiedetty MySQL Community Server 5.0, jonka kehityksestä ja ominaisuuksista on kerrottu aiemmin opinnäytetyön tietokantaohjelmistoa koskevassa teoriaosuudessa.

Tietokannan suunnitteluohjelmistoksi valittiin ilmainen ja GPL-lisensoitu DBDesigner 4. Ohjelman valinnan perusteina oli paitsi sen edullisuus myös helppokäyttöisyys ja mahdollisuus julkaisuvalmiiden raporttien tulostamiseen.

Karhukamera-projektissa suunnitellussa relaatiotietokannassa on 44 taulua ja 51 taulua yhdistävää relaatiota. Relaatiotietokannassa on käytetty MyISAM- ja InnoDB-tietokantamoottoria. Taulut, joilla on jokin relaatioyhteys käyttävät ACID-sääntöjä noudattavaa InnoDB-tietokantamoottoria. Taulut, jotka eivät ole relaatiossa muihin tauluihin, käyttävät nopeampaa MyISAM-tietokantamoottoria. MyISAM-tietokantamoottorin käytöllä koko järjestelmässä olisi saavutettu hieman nopeammat vasteet luku- ja kirjoitusoperaatioissa, mutta InnoDB:n takaama tietokannan viite-ehyden säilyminen oli datan hallinnan kannalta välttämätön ominaisuus.

Sekvenssikuvauksjärjestelmän tietokannan taulut on tietokantasuunnitelmassa jaettu kahteentoista alueeseen, millä on pyritty selventämään taulujen tehtävää järjestelmässä. Esimerkkinä tästä on säähän ja luonnonoloihin liittyvien taulujen keskittäminen WEATHER-alueeseen.

Suunnitellusta relaatiotietokannasta on tehty Karhukamera-projektia varten raportti tietokannan rakenteesta, josta käy ilmi käsitekaavio ja taulujen sisältämät sarakkeet (Liite 1). Lisäksi tietokannasta on laadittu yksilöselosteraportti, jossa on taulujen yksilöselosteet ja sarakkeiden tietotyypit (Liite 2).

## 7 KÄYTTÖLIITTYMÄ

Internet-sivuston suunnittelussa ensimmäinen tehtävä on määrittellä, minkätyyppistä verkkosivustoa ollaan kehittämässä ja mitä ominaisuuksia siltä vaaditaan. Karhukamera.com-sivuston tarkoitus on toimia käyttöliittymänä sekvenssikuvausjärjestelmän tuottaman datan tutkimisessa sekä Karhukamera-projektia esittelevänä yleis-sivustona.

Internet-sivuston asettelulla ja värimaailmalla haettiin ilmeeltään asiallista, selkeää ja helppokäyttöistä kokonaisuutta (Kuvio 3). Sivuston rakenne toteutettiin lisäksi pääte-laitteen näytön mukaan skaalautuvaksi.



Kuvio 3. Karhukamera.com-sivuston etusivu.

### 7.1 Käyttäjääuautikointi

Internet-sivuston käyttäjätunnistukseen käytetään LDAP-protokollaa, joka on hakemistopalveluiden käyttöön tarkoitettu verkkoprotokolla. LDAP-protokolla on yksinkertaistettu versio monimutkaisemmasta X.500-hakemistopalvelusta. Hakemiston

tiedot on järjestetty tavallisesti puuhun organisaatorakenteen mukaisesti ja niitä säilytetään avain-arvopareina. LDAP-pohjaisella autentikointipalvelulla pystyttiin keskittämään kaikkien järjestelmään kuuluvien ohjelmistojen käyttäjätunnistus ja käyttäjätietojen hallinnointi yhteen keskitettyyn palveluun.

Karhukamera-sivustolla käyttäjäautentikoinnista vastaa Kirjautuminen-luokka, jolla on metodit login, logout, ldap\_connection, form\_layout1 ja form\_layout2 (Liite 3). Login-metodi vastaanottaa käyttäjän syötteen, käsittelee ja välittää sen ldap\_connection-metodille (Listaus 2).

```
function login(){
  if(isset($_POST['login'])
  && !empty($_SERVER['HTTP_REFERER'])
  && strpos($_SERVER['HTTP_REFERER'],
  'http://www.karhukamera.com/') === 0){
    $this->username = trim(strip_tags($_POST['username']));
    $this->password = trim(strip_tags($_POST['password']));

    $this->error =
      $this->ldap_connection($this->username, $this->password);

    (strlen($this->error) != 0) ?
      $this->form_layout1($this->error) : $this->form_layout2();
  }

  else {
    (isset($_SESSION['krh_username'])
    && $_SESSION['krh_ip'] === $_SERVER['REMOTE_ADDR']) ?
      $this->form_layout2() : $this->form_layout1(false);
  }
}
```

Listaus 2. Login-metodi.

Ldap\_connection-metodi huolehtii varsinaisesta yhteydenotosta LDAP-palveluun ja käyttäjätunnistuksesta. Metodi saa parametreina käyttäjätunnuksen ja salasanan, joilla se pyrkii kirjautumaan järjestelmään. Kirjautumisen onnistuessa käyttäjä merkitään kirjautuneeksi, muussa tapauksessa palautetaan ilmoitus kirjautumisen epäonnistumisesta (Listaus 3).

```

function ldap_connection($username, $password){
    $this->username = $username;
    $this->password = $password;

    $this->ldaphost = "ldaps://oikeus.karhukamera.com";
    $this->user="uid=".$username.",ou=People,dc=karhukamera,dc=com";
    $this->ldapconn = ldap_connect($this->ldaphost);

    if(!$this->ldapconn)
        return $this->error = ERROR_LDAP_SERVER_NOT_FOUND;

    ldap_set_option($this->ldapconn, LDAP_OPT_PROTOCOL_VERSION, 3);
    $this->bind =
        @ldap_bind($this->ldapconn,$this->user,$this->password);

    if(!$this->bind)
        return $this->error = USERNAME_OR_PASSWORD_INVALID;

    ldap_close($this->ldapconn);

    $_SESSION['krh_username'] = $this->username;
    $_SESSION['krh_ip'] = $_SERVER['REMOTE_ADDR'];
    return $this->error = '';
}

```

Listaus 3. Ldap\_connection-metodi.

Kirjautuessaan ulos järjestelmästä käyttäjä lähettää logout-syötteen, jolloin logout-metodi tuhoaa käyttäjäistunnon (Listaus 4).

```

function logout(){
    if(isset($_POST['logout'])){
        session_unset();
        session_destroy();
    }
}

```

Listaus 4. Logout-metodi.

Luokalla on verkkosivulle kaksi erilaista graafista ilmentymää, jotka näkyvät käyttäjälle kirjautumistilan mukaan. Käyttäjän ollessa järjestelmässä kirjautumattomana, käyttöliittymänä on kirjautumislomake, johon tulostetaan lisäksi virheilmoitus, jos käyttäjän kirjautuminen on epäonnistunut. Kirjautuneelle käyttäjälle käyttöliittymänä on ilmoitus kirjautumisesta järjestelmään ja nappi, josta käyttäjä voi kirjautua ulos. Käyttöliittymän tulostamisesta vastaa metodit form\_layout1 ja form\_layout2, joista



jälkimmäinen huolehtii tulostuksesta silloin kun käyttäjä on kirjautuneena järjestelmään (Kuvio 4).



Kuvio 4. Kirjautumislaatikon ilmentymät kun käyttäjän kirjautuminen on epäonnistunut ja kun käyttäjä on kirjautuneena.

Kirjautuminen-luokkaa ei käytetä verkkosivustolla suoraan vaan käyttöliittymään upotetun login\_block-funktion kautta (Listaus 5).

```
function login_block(){
    $kayttaja = new Kirjautuminen();
    $kayttaja->logout();
    $kayttaja->login();
}
```

Listaus 5. Login\_block-funktio.

## 7.2 Verkkosivuston sivulaatikat

Laatikko-luokka vastaa sivustolla näkyvien pienten sivulaatikoiden tulostamisesta. Laatikko-luokka sisältää metodit sql\_kysely, sql\_vapautus, tulostus\_kommentit, tulostus\_uutiset, tulostus\_mainos ja tulostus\_kuva. Alapuolella näkyvässä esimerkissä käyttöliittymään upotettu news\_block-funktio käyttää Laatikko-luokkaa tulostaakseen viimeisimmät uutiset (Listaus 6).

```

class Laatikko {
    function sql_kysely($sql_query){
        $this->sql_query = $sql_query;
        $this->sql_result = mysql_query($this->sql_query)
            or die('Query failed: ' . mysql_error());
    }

    function sql_vapautus(){
        mysql_free_result($this->sql_result);
    }

    ...

    function tulostus_utiset(){
        while ($this->sql_row = mysql_fetch_assoc($this->sql_result)){
            echo '<div class="post">', "\n";
            echo '<div class="date">'
                . $this->sql_row['pvm'] . '</div>', "\n";
            echo '<div class="topic"><a href="news.php?s='
                . $this->sql_row['id_cat'] . '&nid='
                . $this->sql_row['id'] . '" >'
                . $this->sql_row['asia'] . '</a></div>', "\n";
            echo '</div>', "\n";
        }
        echo '<div class="readmore"><a href="news.php">'
            . READ_OLDER_NEWS . '</a></div>', "\n";
    }
}

function news_block(){
    $utiset = new Laatikko();
    $utiset->sql_kysely("select headline as asia,
        idnews_category as id_cat, modified, idnews as id
        date_format(modified, '%d.%m.%Y %H:%i:%s') as pvm,
        from krh_news order by modified desc limit 5;");
    $utiset->tulostus_utiset();
    $utiset->sql_vapautus();
}

```

Listaus 6. Sivulaatikon tulostaminen.

### 7.3 Tallenteiden hakeminen tietokannasta

Tallenteiden haku tietokannasta on koko järjestelmän tehokkaan käytön kannalta ensiarvoisen tärkeä ominaisuus. Hakulomakkeen suunnittelussa pyrittiin selkeyteen ja luotettavuuteen. Lisäksi virheilmoituksista pyrittiin saamaan käyttäjälle mahdollisimman informatiiviset.

Hakujen nopeutta pyrittiin järjestelmässä parantamaan pääosin kolmella tavalla. Tietokannan indeksoinneilla parannettiin tietokannan vaste- ja hakuajoja. Hakulomakkeen syötteiden käsittelyvaiheessa on erilaisilla käyttäjäkohtaisilla rajoituksilla pienennetty yksittäisen käyttäjän mahdollisuuksia heikentää koko järjestelmän vasteajoja. Tietojenhakuprosessi on lisäksi toteutettu modulaarisesti ja siten, että käyttäjän syötteiden perusteella tehtävät haut voidaan suorittaa mahdollisimman tehokkaassa järjestyksessä.

Luokat Hakulomake, Haku ja Hakutulos, joilla on omat tehtävänsä, vastaavat koko hakuprosessista. Hakulomake-luokka vastaa hakulomakkeen käyttöliittymästä. Hakuluokka on luokista monipuolisin. Se vastaanottaa ja käsittelee käyttäjän syötteet, generoi hakulomakkeelle mahdolliset virheilmoitukset, suorittaa varsinaisen haun tietokannasta ja valmistelea löydetyt tallenteet Hakutulos-luokkaa varten. Hakutulosluokka hakee löytyneet tallenteet tietokannasta ja tulostaa ne käyttäjälle, liittäen tuloksiin vielä tallenteisiin liittyvät kuvaukset.

### 7.3.1 Hakulomake

Hakulomake-luokka tulostaa hakulomakkeen verkkosivulle (Liite 4). Hakulomake muodostuu lomakkeesta, jossa käyttäjä voi valita, millä kriteereillä tallenteita haluaa etsiä. Ensimmäiseksi käyttäjän pitää valita etsikö valokuvia vai videoita. Seuraavaksi käyttäjän pitää valita, minkä kuvauspaikan materiaalista haluaa tallenteita etsiä. Kuvauspaikan valinnan avuksi käyttäjälle ilmoitetaan lomakkeessa milloin eri kuvauspaikat ovat olleet aktiivisia.

Kolmanneksi käyttäjän pitää valita joko aikaväli tai päivämäärät, joilta haluaa tallenteita etsiä. Päivämäärien tai aikavälin valinta tapahtuu projektin ulkopuolella kehitetyllä JavaScript-sovelluksella. Lisäksi käyttäjä voi valita listalta kuvauksen, joka haettavilla tallenteilla pitää olla (Kuvio 5).

: . Haku

**Valitse tiedostotyyppi**

Kuva  
 Video

**Valitse kuvauspaikka**

Ulvila1 
**Kuvauspaikka ( kuvausaika )**  
 Satks1 ( 11.12.2006 - 08.01.2007 )  
 Ulvila1 ( 02.01.2005 - 29.01.2007 )

**Valitse aikaväli**

Aloituspäivämäärä Lopetuspäivämäärä

2007-01-01 19:08:00 [calendar icon] — [calendar icon]

**Tai valitse haluamasi päivämäärät**

Valitut päivämäärät [calendar icon]

**Valitse kuvaukset jotka tiedoston tulee sisältää**

Halo [up]  
 Huolto [list]  
 Ilotulitus [list]  
 Korppi [list]  
 Kuu [list]  
 Lintu [down]

Tammikuu, 2007

Tänään

Wko	Ma	Ti	Ke	To	Pe	La	Su
1	1	2	3	4	5	6	7
2	8	9	10	11	12	13	14
3	15	16	17	18	19	20	21
4	22	23	24	25	26	27	28
5	29	30	31				

Aika **19 : 09**

Valitse päivämäärä

Kuvio 5. Hakulomake.

### 7.3.2 Haku

Haku-luokka on koko hakuprosessin ydinluokka, joka vastaa paitsi itse tietokanta-hausta myös mm. syötteiden vastaanottamisesta ja käsittelystä sekä virheraportoin-

nista (Liite 5). Luokka muodostaa käyttäjän syötteiden perusteella SQL-kyselylauseet ja suorittaa ne arviolta tehokkaimmassa järjestyksessä.

Haku-luokka käyttää tehokkaasti myös kahden muun hakuun liittyvän luokan palveluita ja ohjaa näiden suorituksia. Hyvänä esimerkkinä tästä on `query_null_error`-metodi, joka käyttää ensin ehtolauseessa löydettyjen rivien lukumäärän vertailuun Hakutulos-luokan `query_num_rows`-metodia ja tulostaa ehdon täytyessä Hakulomake-luokan `layout_form`-metodia käyttäen hakulomakkeen virheilmoituksineen (Listaus 7).

```
function query_null_error() {  
    if((Hakutulos::query_num_rows()) < 1){  
        $this->error_array[] = NO_RESULTS;  
        return Hakulomake::layout_form($this->error_array);  
    }  
}
```

Listaus 7. `Query_null_error`-metodi.

### 7.3.3 Hakutulokset

Hakutulos-luokka vastaa hakutulosten listaamisesta käyttäjälle (Liite 6). Käyttäjälle näytetään tulosten otsikko-osassa löydettyjen tallenteiden kokonaismäärä ja mikä osa tuloksista näytetään. Löydettyistä tallenteista tulostetaan käyttäjälle tallenteen nimi, tallenteen päivämäärä, kuvauspaikka ja tallenteeseen liitetyt kuvaukset.

Löydetty tallenteet sivutetaan käyttäjälle 500 tallenteen ryhmiksi, joita selaamalla käyttäjä voi tarkastella kaikkia haulla löydettyjä tallenteita (Kuvio 6).

: . Tulokset 0 - 6 / 6			
Tiedoston nimi	Tiedoston päiväys	Kuvauspaikka	Kuvaukset
<a href="#">20060719_0914.jpg</a>	19.07.2006 09:14:06	Ulvila1	cumulus humilis
<a href="#">20060719_0915.jpg</a>	19.07.2006 09:15:06	Ulvila1	cumulus humilis
<a href="#">20060719_0916.jpg</a>	19.07.2006 09:16:06	Ulvila1	cumulus humilis
<a href="#">20060719_0917.jpg</a>	19.07.2006 09:17:06	Ulvila1	cumulus humilis
<a href="#">20060719_0918.jpg</a>	19.07.2006 09:18:06	Ulvila1	cumulus humilis
<a href="#">20060719_0919.jpg</a>	19.07.2006 09:19:06	Ulvila1	cumulus humilis

◀ Edelliset
[ 1 ]
Seuraavat ▶

Kuvio 6. Hakutulokset.

#### 7.4 Kuvatallenne ja siihen liittyvän informaation esittäminen

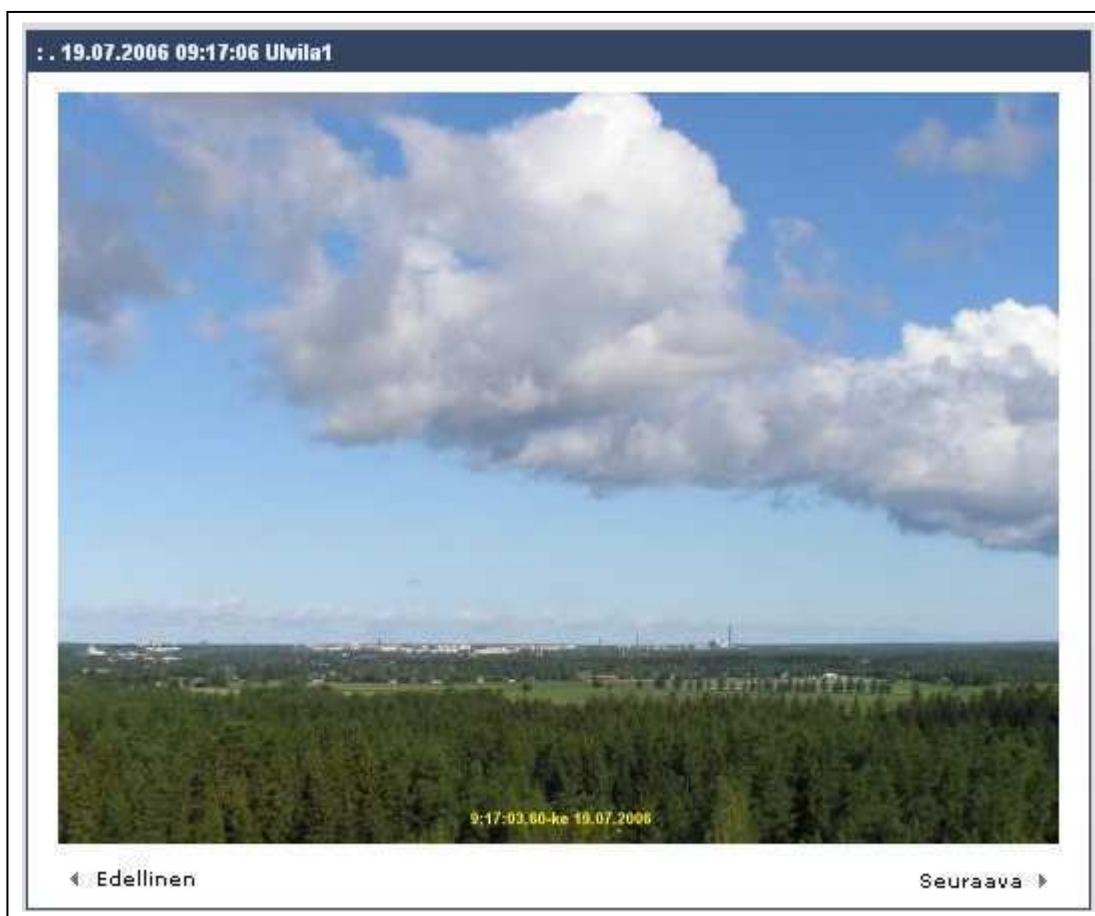
Kuvatallenteen ja siihen yhdistettävän informaation tulostaminen käyttäjälle on hakutoiminnon ohella toinen sekvenssikuvauksjärjestelmän käyttöliittymänä toimimisen edellytys. Kuvatallenne ja siihen yhdistetty informaatio on jaettu erillisiin informaatiolaatikoihin, millä on pyritty järjestelmän käyttäjän kannalta tiedon selkeään jaotteleluun. Kuvatallenteen lisäksi sivulla näytetään siihen liittyvät EXIF-, sää- ja yleis-tiedot. Lisäksi käyttäjälle näytetään kuvatallenteeseen liitetyt kuvaukset. Kirjautunut käyttäjä voi lisätä kuvatallenteeseen myös uusia kuvauksia ja kaikki käyttäjät voivat kommentoida kuvatallenteita vapaamuotoisesti.

Kuvatallenteen ja siihen liittyvän informaation tulostamisesta käyttäjälle huolehtii järjestelmässä Tiedot-luokka (Liite 7).

##### 7.4.1 Kuvatallenne

Kuvatallenne näytetään käyttäjälle omassa informaatiolaatikossaan, jonka otsikosta käy ilmi kuvausaika ja -paikka (Kuvio 7). Kuvatallenteen alapuolella on navigointimahdollisuus edelliseen ja seuraavaan kuvatallenteeseen. Kuvatallennetta napsauttamalla avautuu selaimen uuteen ikkunaan täysikokoinen kuvatallenne. Täysikokoisen kuvatallenteen resoluutio on aina riippuvainen käytetystä kamerasta ja sen asetuksis-

ta, mutta esimerkiksi Ulvilan observatorion katolle sijoitetulla kameralla (Olympus C740UZ) se on 3200x2400 pikseliä (Olympus 2007).



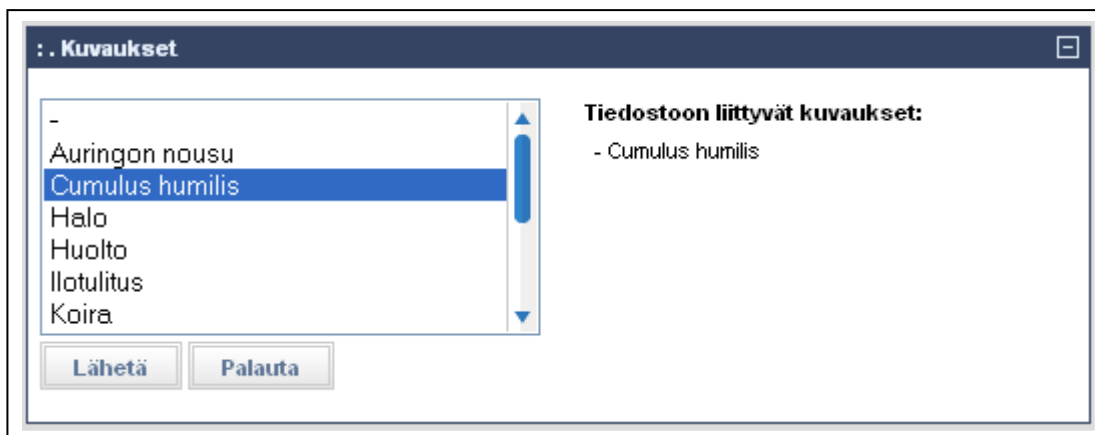
Kuvio 7. Kuvatallenteen näkyminen verkkosivulla.

#### 7.4.2 Kuvaukset

Sekvenssikuvauksjärjestelmän käytölle tutkimukseen on tärkeää, että käyttäjä pystyy luokittelemaan kuvatallenteita niissä näkyvien ominaisuuksien perusteella. Järjestelmässä näistä ominaisuuksista käytetään nimitystä kuvaukset. Tallenteen kuvaukset näyttävällä informaatiolaatikolla on kaksi ilmentymää. Kirjautumaton käyttäjä näkee kuvatallenteeseen liitetyt kuvaukset, mutta ei pysty lisäämään tai poistamaan niitä (Kuvio 8). Kirjautuneella käyttäjällä on mahdollisuus lisätä ja poistaa kuvatallenteeseen liitettyjä kuvauksia (Kuvio 9).



Kuvio 8. Kirjatun käyttäjän kuvaukset.



Kuvio 9. Kirjautuneen käyttäjän kuvaukset.

Hakuvaiheessa käyttäjällä on mahdollisuus hakea tallenteita aiemmin määriteltyjen kuvausten perusteella.

#### 7.4.3 EXIF-tiedot

Digitaalikamerat tallentavat kuvatiedostoon tekstimuotoista oheisinformaatiota. Oheisinformaation tallentamiseen käytetään EXIF-standardia, joka tallentaa kuvaan teknisiä tietoja mm. valotuksesta ja kuvausajankohdasta. EXIF-tiedot tarjoavat käyttäjälle mahdollisuuden arvioida kuvatallennetta myös sen teknisten tietojen perusteella (Kuvio 10).



: . EXIF-tiedot			
Tiedoston nimi	: 20060719_0917.JPG	Valotusaika	: 0.0020 s
Tiedoston koko	: 240 kB	Valotustapa	: Automaattinen valotus
Päivämäärä	: 19.07.2006 09:17:06	Aukkosuhde	: f/8
Kameran valmistaja	: Olympus Corporation	ISO-nopeus	: 200
Kameramalli	: C740UZ	Valkotasapaino	: Käsinsäädetty valkotasapaino
Resoluutio	: 1280x960	Valonlähde	: Hyvä sää
Pottoväli	: 6.8000 mm	Mittaustapa	: Kuvio

Kuvio 10. EXIF-tiedot.

Kuvatallenteen EXIF-tiedot, jotka ovat tallennettuna tietokantaan, tulostaa Tiedotluokan `show_exif`-metodi. Metodi käyttää tietojen hakemiseen tietokantaan tallennettua proseduuria `get_exif_data`, joka kokoaa EXIF-tiedot valmiiksi kutsuvalle metodille (Liite 8). Metodi kutsuu proseduuria ja välittää sille parametreina kuvatallenteen tunnusteen ja käytettävän kielen (Listaus 8). Proseduuria hakee tiedot ja tallentaa ne tietokantaan muuttujiksi metodin haettavaksi.

```
mysql_query("CALL get_exif_data($this->item, $this->language,
    @idimage, @nfsurl, @filename, @filesize, @date, @cameramadeby,
    @cameramodel, @resolution, @isoequiv, @aperture, @exposuretime,
    @focallenght, @exposuremode, @lightsource, @meteringmode,
    @whitebalance);");

$result = mysql_query("SELECT @idimage, @nfsurl, @filename,
    @filesize, @date, @cameramadeby, @cameramodel, @resolution,
    @isoequiv, @aperture, @exposuretime, @focallenght,
    @exposuremode, @lightsource, @meteringmode, @whitebalance;");

$row = mysql_fetch_assoc($result);
```

Listaus 8. Exif-tietojen hakeminen tietokannasta.

#### 7.4.4 Sää tiedot

Kuvatallenteiden tulkinnessa on tärkeää tietää kuvaushetken sää. Nykyisessä järjestelmässä sää tiedot haetaan kuvauspaikkaa lähimpänä sijaitsevasta METAR-dataa julkaisevasta säämittausasemasta. Koska järjestelmällä ei ole omaa sääasemaa, joka takaisi sää tietojen säännöllisen ja luotettavan saamisen, käyttäjälle ilmoitetaan lisätietona vielä arvioitu voimassaoloaika näytettävillä sää tiedoille (Kuvio 11).

: Säätila			
Lämpötila	: 14 °C	Ilmankosteus	: 67 %
Tuulensuunta	: Luode	Ilmanpaine	: 1017.9 mb
Tuulennopeus	: 6.1691 m/s	Puuskissa	: 0 m/s
Kastepiste	: 8 °C	Sää	: Puolipilvistä
Sademäärä	: 0 mm	Tapahtuma	:
Näkyvyys	: 9978 m	Kelpoisuusaika	: 19.07.2006 09:05:00 - 19.07.2006 09:35:00

Kuvio 11. Säätiiedot.

Säätietöjen tulostamisesta käyttäjälle vastaa Tiedot-luokan `show_weather`-metodi (Liite 7). Säätietöjä tulostettaessa haetaan ensin tietokannasta kuvatallenteelle sopivimman säädatan id-tunniste `get_weather_id`-metodilla (Listaus 9). Metodi suorittaa ensin kyselyn tarkalla aikavälillä ja jos ajankohdalta ei löydy voimassaolevaa säädätää, metodi kutsuu tietokantaan tallennettua proseduuria `get_weather_id`, joka palauttaa lähimmän kuvatallenteelle sopivan säädatan id-tunnisteen (Liite 8).

```
function get_weather_id(){
    $result = mysql_query("SELECT idweather as weather
FROM krh_weather w, krh_imageinfo i
WHERE i.filedate < w.validtime_end
AND i.filedate > w.validtime_begin
AND i.idimage = $this->item LIMIT 1;");

    if(mysql_num_rows($result) != 0) {
        $weather = mysql_fetch_assoc($result);
        mysql_free_result($result);
        return $weather['weather'];
    }
    else {
        mysql_query("CALL get_weather_id($this->item, @id_weather);");
        $result = mysql_query("SELECT @id_weather;");
        $weather = mysql_fetch_assoc($result);
        mysql_free_result($result);
        return $weather['@id_weather'];
    }
}
```

Listaus 9. Kuvaan sopivan säädatan id-tunnisteen hakeminen.

Säätietöjen hakemiseen tietokannasta käytetään tietokantaan tallennettua `get_weather_data`-proseduuria, jolle välitetään parametreina halutun säädatan id-tunniste ja käytettävä kieli. Proseduuria suorittaa kyselyn tietokannassa ja tallentaa sääti-

dot muuttujiin, jotka show\_weather\_data-metodi hakee ja tallentaa taulukkoon tulostamista varten (Listaus 10).

```
$this->idweather = $this->get_weather_id();

mysql_query("CALL get_weather_data($this->idweather,
$this->language, @temperature, @windspeed, @dewpoint,
@visibility, @precipitation, @humidity, @pressure,
@gust_speed, @valid, @winddirection, @conditions, @events);");

$result = mysql_query("SELECT @temperature, @windspeed,
@dewpoint, @visibility, @precipitation, @humidity, @pressure,
@gust_speed, @valid, @winddirection, @conditions, @events;");

$row = mysql_fetch_assoc($result);
```

Listaus 10. SQL-lauseet, joilla kutsutaan get\_weather\_data-proseduuria ja haetaan proseduurin tallentamat tiedot taulukkoon.

#### 7.4.5 Muut tiedot

Tiedot, jotka eivät ole kuvatallenteiden tulkinnan kannalta olennaisia, on koottu omaan informaatiolaatikkoon. Informaatiolaatikko sisältää mm. tietoja kuvauspaikasta, kamerasta ja säämittausasemasta (Kuvio 12). Lisäksi käyttäjälle tarjotaan pikahaku videoihin, joihin tarkasteltava kuvatallenne kuuluu. Tietojen tulostamisesta vastaa Tiedot-luokan show\_other\_info-metodi (Liite 7).

: Muut tiedot	
<b>Kuvauspaikka</b>	
Nimi : Ulvila1	Sijainti : Ulvila, Finland
Vastuuhenkilö : <a href="#">mikko</a>	Postinumero : 28450
Kuvausaika : 02.01.2005 -	Gps : DD (61.433-21.9306)
Kuvaussuunta : 313	Korkeus merenpinnasta : 49 m
<p>Ulvilan tähtitornilla on kiinteästi asennettuna yksi kamera. Kamera on suljetussa, ruostumattomasta teräksestä valmistetussa kotelossa jossa on 80mm UV linssi. Kameraa ja kameran koteloa lämmittää pieni alumiinilevyjen väliin sijoitettu vastusmatt</p>	
<b>Kamera</b>	
Valmistaja : Olympus	Kennon tyyppi : CCD
Malli : C-740 UltraZoom	Mpx : 3.2
Lisätietoja : <a href="#">Linkki</a>	
<b>Säämittausasema</b>	
Paikka : Porin lentokenttä	
Sijainti : <a href="#">Pori, Finland</a>	
ICAO : <a href="#">EFPO</a>	
<b>Video</b>	
Onko kuva jossain videossa? : <a href="#">Haku</a>	

Kuvio 12. Muut tiedot.

## 8 VIDEOT KUVATALLENTEIDEN KÄYTÖN TUKENA

Konenäön sijasta sekvenssikuvauksjärjestelmän tuottaman valtavan kuvatallemäärän läpikäyntiin käytetään järjestelmässä videotekniikkaa, jossa valitut kuvatalenteet koostetaan yhdeksi videoksi, jolloin katsojalle syntyy vaikutelma ns. nopeutetusta videosta. Katsojan on helppo ja nopea käydä läpi suuriakin kuvatallemääriä etsiessään kiinnostavia ilmiöitä tai tapahtumia. Video voi myös auttaa ymmärtämään pidempiaikaisia ilmiöitä yksittäisten kuvatalenteiden selaamista paremmin. Kuvatalenteista koostetaan tällä hetkellä vuorokausi-, kuukausi- ja vuosikohtaiset videot.

Sekvenssikuvauksjärjestelmän tietokantaan on videoiden tekemistä varten määritelty taulut `krh_collections` ja `krh_encodeprofile`. `Krh_collections`-taulussa on nimettyinä

muun muassa kokoelman nimi ja kuvaus. Krh\_encodeprofile-taulussa on kuvatallenteista tehtävälle videolle määriteltynä tekniset tiedot, joiden mukaan video koostetaan (Liite 2).

Videot tehdään ilmaisella ja avoimeen lähdekoodiin perustuvalla MPlayer-media-soitinohjelman MEncoder-laajennuksella. MEncoder on komentoriviltä suoritettava ohjelma, jolle PHP-skripti välittää tietokannasta haetut tiedot parametreina. Kun video valmistuu, sama PHP-skripti tallentaa sen tiedot tietokantaan. (Rautanen 7.3.2007.)

Video näytetään käyttäjälle verkkosivulla samalla tavalla kuin yksittäinen kuvatallenne. Video avataan kuitenkin todellisuudessa omalle verkkosivulleen ja näytetään käyttäjälle iframe-kehyksessä. Koska videoiden tarkoituksena on toimia vain tukena kuvatallenteiden käytölle, niistä ei tarjota käyttäjälle kuin teknisiä perustietoja kuten resoluutio ja tiedostokoko. Lisäksi käyttäjä voi kommentoida videota vapaamuotoisesti.

## 9 VARMUUSKOPIOINTI

Karhukamera-projektin käytössä olevalla palvelimella on kahdeksan kiintolevyä, joilla on yhteensä noin 1,7 teratavua tallennuskapasiteettia. Kiintolevyt on liitetty kahteen erilliseen RAID-levyjärjestelmään. RAID-levyjärjestelmät yhdistävät yksittäiset fyysiset kovalevyt yhdeksi loogiseksi kokonaisuudeksi, jonka tavoitteena on parantaa järjestelmän vikasietoisuutta ja nopeutta.

Kaksi kiintolevyä on yhdistetty RAID1-levyjärjestelmään, jossa kiintolevyt peilaavat datan toisilleen, jolloin toisen levyn hajotessa data säilyy kuitenkin edelleen toisella kiintolevyllä. Kuusi muuta kiintolevyä on liitetty RAID5-levyjärjestelmään, jossa pariteettidataa käyttämällä saavutetaan paitsi yhden kiintolevyn hajoamisen kestävä vikasietoisuus myös nopeampi datan lukunopeus. Kumpikaan käytössä oleva levyjär-

jestelmä ei kuitenkaan kestä kahden kiintolevyn samanaikaista hajoamista vaan tällöin menetetään kaikki järjestelmässä hajoamishetkellä oleva data.

Karhukamera-projektin käytössä olevalla palvelimen datan varmuustallennus on toteutettu ilmaisella ja avoimen lähdekoodin Bacula-ohjelmistolla. RAID1-levyjärjestelmässä olevalle järjestelmä- ja tietokantadatalle suoritetaan kerran viikossa ns. täysvarmistus RAID5-levyjärjestelmään. Lisäksi päivittäin suoritetaan inkrementaali-varmistus eli varmistetaan viimeisen täysvarmistuksen jälkeen muuttuneet tiedot. Varmuustallennuksista säilytetään aina kaksi viimeisintä versiota. (Rautanen 7.3.2007.)

Sekvenssikuvauksessa syntyvät tallenteet säilytetään Karhukamera-projektin käytössä olevan palvelimen RAID5-levyjärjestelmässä. Tallenteet poltetaan lisäksi kahden eri valmistajan laadukkaille DVD-R-levyille ja levyt säilytetään huolellisesti erillään ja pystyasennossa. Levyt on lisäksi suojattu hyvin valolta ja mekaaniselta rasitukselta.

## 10 PROJEKTIN NYKYTILA JA TULEVAISUUS

### 10.1 Projektin nykytila

Sekvenssikuvauksjärjestelmää on kehitetty tietotekniikan ja valokuvauksen harrastajien ja ammattilaisten toimesta vuodesta 2004. Projektiin on osallistunut 17 henkilöä. Järjestelmässä on käytetty kolmea kameraa, joista Ulvilan observatorion katolle sijoitettu kamera saavutti miljoonan tietokantaan tallennetun kuvan rajapyykin 10.1.2007. Toinen kamera on tarkoitettu palvelemaan Ulvilan tähtitornilla, erilaisten taivaan ilmiöiden kartoittajana. Kolmas kamera, MUU eli Mostly Unidentified Unit, liikkuu omin avuin ja pystyy kuvaamaan akkujen voimalla noin 12 tuntia. (Viitapohja 2007b.)

Tähtitieteellinen yhdistys Ursa ry myönsi joulukuussa 2006 sekvenssikuvauusjärjestelmän jatkokehittämistä varten ns. harrastaja-apurahan.

## 10.2 Projektin tulevaisuus

Sekvenssikuvauusjärjestelmää tullaan kehittämään jatkossa edelleen yritysten, yhteisöjen ja säätiöiden tuella havaintovälineeksi, jota voitaisiin hyödyntää tutkimuksessa, jossa visuaalisella havainnolla on merkittävä asema. Sekvenssikuvauusjärjestelmä voi tulevaisuudessa olla tehokas, monitieteellinen työkalu erilaisten visuaalisten havaintojen tuottamiseen ja tukemiseen. (Viitapohja 2007a.)

Tulevaisuuden kehityskohteina on suunniteltu erityisen liikkuvan yksikön kehittämistä, jolla tutkittaisiin GPS-datan yhdistämistä tietokantaan sekä merentutkimuksen alueelta satelliittidatan käyttämistä havaintovälinejärjestelmän osana. Kiinnostava kehityskohde on myös ornitologian käyttöön kehitettävä järjestelmä, jossa visuaalista havaintoa tukemaan liitettäisiin ääninäyte.

## 10.3 Internet-sivuston kehittäminen tulevaisuudessa

Internet-sivuston kehittäminen kokonaiseksi ja monipuoliseksi julkaisujärjestelmäksi on osoittautunut huomattavasti arvioitua työläemmäksi tavoitteeksi ja siihen käytetty aika on ollut poissa projektin varsinaisten ydinalueiden kehittämiseltä. Alustavasti on suunniteltu luovuttavan omasta julkaisujärjestelmästä ja siirryttävän käyttämään jo olemassa olevaa avoimen lähdekoodin julkaisujärjestelmää.

Valmista julkaisujärjestelmää hyödyntämällä Karhukamera-projektissa voidaan keskittyä kehittämään järjestelmän kannalta tärkeitä komponentteja, kokonaisen julkaisujärjestelmän sijaan. Komponenttien kehittämisellä saavutetaan myös erinomainen julkaisujärjestelmän projektikohtainen muokattavuus, kun järjestelmään voidaan implementoida juuri tutkimuskohteen tarvitsemat komponentit.

Käyttöliittymän seuraavassa kehitysversiossa tullaan lisäksi käyttämään AJAX-tekniikkaa, jolla verkkosovelluksesta pystytään kehittämään enemmän perinteisen työpöytäsovelluksen kaltainen ja lisäämään verkkosovelluksen vuorovaikutteisuutta, nopeutta ja käytettävyyttä. AJAX-tekniikkaa käyttämällä verkkosivua ei tarvitse ladata aina uudelleen vaan selvittää päivittämällä yksittäinen verkkosovelluksen komponentti.

#### 10.4 Relaatiotietokannan kehittäminen tulevaisuudessa

Relaatiotietokanta on jo tällä hetkellä hyvin monipuolinen ja siinä on varauduttu moniin tulevaisuudessa tarvittaviin ominaisuuksiin. Tulevaisuudessa muutoksia tietokantaan tulee uusien tutkimuskohteiden vaatimuksista ja näiden vaatimusten implementoinnista havaintovälinejärjestelmään.

Julkaisujärjestelmän vaihtaminen tulee lisäksi aiheuttamaan sen, että tietokannasta voidaan poistaa tauluja, joiden ominaisuudet on jo toteutettu julkaisujärjestelmän omassa tietokannassa.

## 11 YHTEENVETO

Opinnäytetyön tavoitteena oli kehittää relaatiotietokanta uudentyyppiselle sekvenssi-kuvausta käyttävällä havaintovälineelle. Lisäksi järjestelmälle oli suunniteltava ja toteutettava Internet-sivusto, jonka tuli toimia paitsi käyttöliittymänä sekvenssi-kuvausjärjestelmän tuottamaan materiaaliin, myös esitellä Karhukamera-projektin tavoitteita ja etenemistä.

Opinnäytetyötä kehitettiin tiiviissä yhteistyössä Karhukamera-projektiryhmän kanssa, jolta saatu palaute otettiin suunnittelussa jo varhain huomioon. Erityisen suuri vaikutus projektiryhmän palautteella oli relaatiotietokannan suunnittelussa.



Opinnäytetyön tekeminen syvensi osaamistani niin PHP-komentosarjakielen kuin MySQL-tietokantaohjelmiston ominaisuuksista ja mahdollisuuksista. Erityisen mielenkiintoisiksi koin MySQL 5.0 -version uudet ominaisuudet, kuten näkymät ja tallennettavat proseduurit, joita käytinkin runsaasti opinnäytetyössäni.

Opinnäytetyön tekeminen osoitti myös miten tärkeää on tavoitteen tarkka määrittely. Karhukamera-projektissa kehitettiin aivan uudenlaista järjestelmää ja sille asetetut tavoitteet kasvoivat projektin edetessä. Järjestelmän ominaisuuksien lisääntyminen aiheutti luonnollisesti myös opinnäytetyöhön kuluneen ajan kasvamista ja näin myös aikataulun venymistä. Jälkeenpäin tarkastellessa ainakin osa kehitystyön aikana tulleista muutoksista olisi voitu välttää paremmalla tavoitteiden ja ominaisuuksien suunnittelulla projektin alussa.

Opinnäytetyönä suunniteltu relaatiotietokanta on todettu käytössä toimivaksi ja sen ominaisuudet tulevat riittämään pitkälle myös projektin tulevissa hankkeissa. Internet-sivusto tarjoaa toteutetulle sekvenssikuvauksjärjestelmälle selkeän ja toimivan käyttöliittymän, jolla käyttäjä pystyy tehokkaasti hyödyntämään järjestelmään tallennettua informaatiota. Uskon niin relaatiotietokannan kuin Internet-sivustonkin tarjoavan hyvän perustan järjestelmän kehittämiseksi myös tulevaisuudessa.

## LÄHTEET

Axelsson J., Mark Birbeck M. & Dubinko M. 2006. XHTML 2.0 [verkkodokumentti]. World Wide Web Consortium. [Viitattu 14.3.2007]. Saatavissa: <http://www.w3.org/TR/xhtml2/xhtml2.pdf>

Boumphrey F., Greer C. & Raggett D. 2000. INSIDE XHTML Ohjelmoijan käsikirja. Helsinki. Oy Edita Ab

Gray, J. 1981. The Transaction Concept: Virtues and Limitations [verkkodokumentti]. Cupertino, CA. Tandem Computers Incorporated. [Viitattu: 18.12.2006]. Saatavissa: <http://research.microsoft.com/~gray/papers/theTransactionConcept.pdf>

Heinäsuo R. 2003. PHP ja MySQL Tietokantapohjaiset verkkopalvelut. Helsinki. Talentum Media Oy

MySQL AB. 2006. MySQL 5.0 Reference Manual [verkkodokumentti]. MySQL AB. [Viitattu 15.12.2006]. Saatavissa: <http://downloads.mysql.com/docs/refman-5.0-en.a4.pdf>

Netcraft Ltd. 2006. December 2006 Web Server Survey [verkkodokumentti]. Netcraft Ltd. [Viitattu: 3.1.2007]. Saatavissa: [http://news.netcraft.com/archives/2006/12/05/december\\_2006\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2006/12/05/december_2006_web_server_survey.html)

Olympus. 2007. Specifications for C-740 UltraZoom [verkkodokumentti]. Olympus Corporation. [viitattu 15.3.2007]. Saatavissa: [http://www.olympus.fi/consumer/29\\_C-740\\_UltraZoom\\_Specifications.htm](http://www.olympus.fi/consumer/29_C-740_UltraZoom_Specifications.htm)

Suraski, Z. 2004. Changes in PHP 5 / Zend Engine II [verkkodokumentti]. Zend Technologies Ltd. [Viitattu: 3.1.2007]. Saatavissa: <http://www.zend.com/php5/articles/engine2-php5-changes.php>

PHP Documentation Group. 2007a. History of PHP and related projects [verkkodokumentti]. The PHP Documentation Group. [Viitattu 3.1.2007]. Saatavissa: <http://www.php.net/manual/en/history.php>

PHP Documentation Group. 2007b. PHP Manual [verkkodokumentti]. The PHP Documentation Group. [Viitattu 3.1.2007]. Saatavissa: <http://www.php.net/manual/en/>

Rautanen O. 2007. Karhukamera, Tampere. Verkkohaastattelu 7.3.2007. Haastattelija Teemu Jönkkäri. Muistiinpanot haastattelijan hallussa.

Viitapohja M. 2007a. Karhukamera, Pori. Verkkohaastattelu 10.3.2007. Haastattelija Teemu Jönkkäri. Muistiinpanot haastattelijan hallussa.

Viitapohja M. 2007b. Sekvenssikuvausjärjestelmän esittely [verkkodokumentti]. Karhukamera. [Viitattu: 28.2.2007]. Saatavissa: [http://www.karhukamera.com/project/docs/Karhukamera\\_esittely.pdf](http://www.karhukamera.com/project/docs/Karhukamera_esittely.pdf)

Wikipedia. 2007a. MySQL AB [verkkodokumentti]. Wikimedia Foundation Inc. [Viitattu: 14.3.2007]. Saatavissa: [http://en.wikipedia.org/wiki/MySQL\\_AB](http://en.wikipedia.org/wiki/MySQL_AB)

Wikipedia. 2007b. ACID [verkkodokumentti]. Wikimedia Foundation Inc. [Viitattu: 18.12.2006]. Saatavissa: <http://en.wikipedia.org/wiki/ACID>

Wikipedia. 2007c. Acid2 [verkkodokumentti].Wikimedia Foundation Inc. [Viitattu: 10.1.2007]. Saatavissa: <http://en.wikipedia.org/wiki/Acid2>

Wikipedia. 2007d. Cascading Style Sheets [verkkodokumentti].Wikipedia Foundation Inc. [Viitattu: 10.1.2007]. Saatavissa: [http://en.wikipedia.org/wiki/Cascading\\_Style\\_Sheets](http://en.wikipedia.org/wiki/Cascading_Style_Sheets)