



# **Kykylaakso-ohje ylläpitäjälle**

Mika Laikko

Opinnäytetyö  
Toukokuu 2014  
Tietotekniikka  
Ohjelmistotekniikka

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietotekniikan koulutusohjelma  
Ohjelmistotekniikan suuntautumisvaihtoehto

Mika Laikko:  
Kykylaakso-ohje ylläpitäjälle

Opinnäytetyö 45 sivua, joista liitteitä 3 sivua  
Toukokuu 2014

---

Tämän työn tarkoituksena on olla ohjeena Kykylaaksojärjestelmän ylläpitäjälle. Tässä dokumentissa käydään läpi järjestelmän toteutuksessa käytettyjä ohjelmointikieliä ja niihin liittyviä tekniikoita. Tähän dokumenttiin sisältyy myös selostusta Kykylaakson arkkitehtuurista ja teknistä dokumentaatiota. Työssä käydään myös läpi perusteita relaatiotietokannoista ja SQL-kyselyistä. Kykylaakso järjestelmää ylläpidetään Debian palvelimella ja tämän vuoksi käydään läpi hieman linux-käyttöjärjestelmän käyttöä.

---

Asiasanat: järjestelmä dokumentaatio, tekninen selvitys, toiminta analyysi

## **ABSTRACT**

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Information Technology  
Software engineering

**MIKA LAIKKO:**  
Kykylaakso manual for administrator

Bachelor's thesis 45 pages, appendices 3 pages  
May 2014

---

Purpose of this thesis is to be a manual for Kykylaakso system administrator. In this document there will be explanations of the programming languages and techniques which are being used in the system. Also there will be information about the architecture and technical documentation of the system on high level. Relational databases and SQL-queries are also explained on this document. The Kykylaakso system runs on Debian server and for that there will be some guides for working in Linux environment.

---

Key words: system documentation, technical debriefing, action analysis

## SISÄLLYS

1	JOHDANTO.....	7
2	JÄRJESTELMÄN TOTEUTUKSESSA KÄYTETYT TEKNIIKAT.....	8
2.1	HTML: HyperText Markup Language .....	8
2.1.1	HTML historia .....	8
2.1.2	HTML-syntaksi .....	8
2.2	PHP: Hypertext Preprocessor .....	9
2.2.1	Historia.....	10
2.2.2	PHP-syntaksi .....	11
2.2.3	PHP-istunto .....	13
2.2.4	URL-parametrit.....	14
2.2.5	Evästeet .....	15
2.3	JavaScript.....	16
2.3.1	Kirjastojen käyttö verrattuna natiiviin JavaScriptiin.....	16
3	TIETOKANTA .....	18
3.1	Relaatiotietokanta .....	18
3.1.1	Yhteydet .....	19
3.2	MySQL .....	21
3.3	SQL.....	21
3.3.1	Syntaksi.....	22
3.4	Kykylaakson tietokanta.....	25
4	PALVELINYMPÄRISTÖ .....	26
4.1	Debian GNU/Linux.....	26
4.1.1	Palvelimeen yhdistäminen .....	27
4.1.2	Hyödyllisiä komentoja .....	28
5	OHJELMISTOARKKITEHTUURI.....	33
5.1	Kykylaakson arkkitehtuuri.....	34
5.1.1	Kykylaakso tekninen dokumentaatio .....	34
5.1.2	Pupesoft.....	35
5.1.3	TAMK:n osat .....	37
5.1.4	Ajoitetut tehtävät.....	38
6	YHTEENVETO .....	40
	LÄHTEET.....	41
	LIITTEET .....	43
	Liite 1. Kykylaakson järjestelmäkaavio (Kykylaakso dokumentit: Malli).....	43

## LYHENTEET JA TERMIT

AJAX	Asynchronous JavaScript And XML on tekniikka, jolla selainohjelma vaihtaa pieniä määriä dataa palvelimen kanssa taustalla niin, ettei koko verkkosivua tarvitse ladata uudelleen.
C#	Microsoft-yhtiön .NET-konseptia varten kehittämä ohjelmointikieli, joka julkaistiin kesäkuussa 2000.
CERN	Euroopan hiukkasfysiikan tutkimuskeskus. Lyhennys tulee sanoista Conseil Européen pour la Recherche Nucléaire.
CGI	Common Gateway Interface on tärkeä Web-ympäristön tekniikka, jonka avulla selain voi välittää dataa palvelimella suoritettavalle ohjelmalla.
Debian	Suosittu GNU/Linux jakelupaketti.
GNU	”GNU’s Not Unix” on projekti, jonka tavoitteena on kehittää täysin vapaa käyttöjärjestelmä.
ERP	Enterprise resource planning eli toiminnanohjausjärjestelmä on yrityksen tietojärjestelmä, joka integroi eri toimintoja kuten tuotantoa, jakelua ja varastonhallintaa.
HTML	Hypertext Markup Language on sivunkuvauskieli, jota käytetään web-sivujen sisällön kuvaukseen.
JavaScript	Web-ympäristössä käytettävä komentosarjakieli.
jQuery	Avoimen lähdekoodin JavaScript-kirjasto.
Kernel	Ydin joka määrittelee käyttöjärjestelmän rakenteen, luokituksen ja ominaisuudet.
Linux	Linux-ydintä käyttävien Unixin kaltaisten käyttöjärjestelmien perhe.
MVC	Model-View-Controller on nykyaikana yksi yleisimmin käytettyjä web-arkkitehtuureja.
MVVM	Model-View-ViewModel on Microsoftilta alkunsa saanut ohjelmistoarkkitehtuuri.
MVW	Model-View-Whatever on ohjelmistoarkkitehtuuri, jota toteuttaa esimerkiksi ohjelmistokomponenttikirjasto AngularJs.

PHP	Hypertext Preprocessor on Perlin kaltainen ohjelmointikieli, jota käytetään erityisesti web-palvelinympäristöissä.
Pupesoft	Devlabin kehittämä avoimen lähdekoodin toiminnanohjausjärjestelmä.
SQL	Structured query language on standartoitu kyselykieli, jolla voidaan tehdä hakuja, muutoksia ja lisäyksiä relaatiotietokantaan.
SSH	Secure Shell on salattuun tietoliikenteeseen tarkoitettu protokolla, jonka yleisin käyttätapa on ottaa etäyhteys SSH-palvelimeen merkkipohjaisen konsolin kautta.
TAMK	Tampereen ammattikorkeakoulu
yacc	Yet another compiler compiler on jäsentäjä ja jäsentäjä-generaattori. Sen mahdollistaa nopean ja tehokkaan parserien luomisen mielivaltaisen monimutkaisille kieliopille.

## 1 JOHDANTO

Kykylaakso on tampereen ammattikorkeakoulun virtuaalinen oppimisympäristö jossa tradenomiopiskelijat harjoittavat liiketoimintaa ilman siihen liittyviä tosielämän riskejä.

Kykylaakso-järjestelmä rakentuu useasta palasesta, jotka ovat yhteydessä toisiinsa erillisillä rajapintakomponenteilla. Näitä palasia ovat Devlab:n kehittämä ERP-järjestelmä pupesoft, pankki-, vero- ja verkkokauppa-sovellukset.

Järjestelmän alkuperäisistä palasista, eikä niihin tehdyistä muutoksista ei ole olemassa dokumentaatiota. Eikä kukaan tiedä miten tai miksi asiat toimivat tai eivät toimi. Tämän dokumentaation tarkoitus on valoittaa järjestelmän toimintaa tuleville ylläpitäjille ja kehittäjille, jotta heidän ei tarvitsisi tutkia jotain mikä on jo tutkittu.

## 2 JÄRJESTELMÄN TOTEUTUKSESSA KÄYTETYT TEKNIIKAT

### 2.1 HTML: HyperText Markup Language

HTML on sivunkuvauskieli, jolla voidaan kuvata WWW-sivujen rakenne ja ulkoasu. HTML:n avulla voidaan sivu jakaa osioihin kuten otsakkeisiin, palstoihin ja tauluihin. Ehkä tärkein ominaisuus HTML:ssa on hypertekstin kuvaus. Hyperteksti mahdollistaa automaattiset hyperlinkeiksi kutsutut ristiviittaukset eri dokumenttien välillä. HTML-dokumentti voidaan luoda millä tahansa tekstieditorilla tallentamalla tiedosto .html-päätteellä. HTML-dokumentti voi olla staattinen eli etukäteen valmiiksi tehty tai dynaaminen, jolloin se muuttuu käytön yhteydessä. HTML:ää voidaan generoida dynaamisesti ohjelmointikielillä kuten PHP, C# tai skriptikielillä kuten JavaScript. (The World Wide Web Consortium 2014.)

#### 2.1.1 HTML historia

HTML:n historia alkoi jo vuonna 1989, jolloin Tim Berners-Lee työskennellessään CERNissä kirjoitti ehdotelman Internetiä hyödyntävästä hyperteksti- järjestelmästä, joka voisi korvata CERNin käytössä olevan kirjavan dokumenttiformaatin. Eli alunperin HTML kehitettiin helpottamaan CERNissä työskentelevien tiedemiesten välistä kommunikointia. Berners-Lee määrittäi ensimmäisen HTML-version ja kirjoitti ensimmäiset selain- ja palvelinohjelmistot vuoden 1991 loppuun mennessä, jolloin CERN käynnisti oman WWW-palvelimensa. Monet ideasta kiinnostuneet kuten Robert Caillau, liittyivät projektiin nopeasti. (The World Wide Web Consortium: (The World Wide Web Consortium 2014.)

#### 2.1.2 HTML-syntaksi

HTML on rakenteista tekstiä, joka muodostuu sisäkkäisistä ja peräkkäisistä elementeistä. Elementtejä edustavat kulmasulkein merkityt tunnisteet. Selain ei näytä tunnisteita, vaan käsittelee niitä teknisinä ohjeina. Näin selain jäsentää sivun varsinaisen sisällön, joka näytetään käyttäjälle. (The World Wide Web Consortium 2014.)



Tavallisesti elementeissä on erikseen aloitustunniste ja vinoviivalla merkitty lopetustunniste. Tunnisteiden väliin jää sisältö jossa voi olla tekstiä ja muita HTML-elementtejä. Vain tunnisteiden väliin jäävä osio näkyy käyttäjälle selaimessa. Aloitustunnisteeseen saattaa myös sisältyä attribuutteja, jotka tarkemmin määrittelevät elementin ominaisuuksia. Elementit ja niiden attribuutit ovat ennalta määrättyjä HTML-standardissa. (The World Wide Web Consortium 2014.)

Esimerkissä 1. on kuvattu toimiva HTML-elementti. Aloitustunniste `<a>` merkitsee hyperlinkkiä. Aloitustunnisteen sisällä annetaan hyperlinkille attribuutti `href` joka määrittää linkin kohteen, eli tässä tapauksessa TAMKin verkkosivujen URL-osoitteen. Tunnisteiden väliin jäävä osuus eli merkkijono ”TAMK” näkyy käyttäjän selaimessa linkkinä josta painamalla pääsee TAMKin verkkosivuille. Linkin kohde voi olla myös paikallinen html-dokumentti, jolloin siihen viitataan tiedostopolulla, joka johtaa kyseiseen tiedostoon.

```
<a href="http://www.tamk.fi">TAMK</a>
```

ESIMERKKI 1. HTML-hyperlinkki

## 2.2 PHP: Hypertext Preprocessor

PHP on laajalti käytetty avoimen lähdekoodin skriptaus kieli, jota käytetään erityisesti web-palvelinympäristöissä dynaamisten web-sivujen luomisessa. PHP on komentosarjakieli eli ohjelmakoodi tulkitaan suoritusvaiheessa, eikä sitä tarvitse kääntää niin kuin C-kieliä. Kielen syntaksi muistuttaa hyvin paljon C:n ja Perl:n syntaksia. Nykypäivänä PHP tukee olio-ohjelmointia ja sisältää monipuolisen tuen luokille, funktioille, abstrakteille luokille, luokkien periytymiselle ja poikkeusten käsittelylle. Kielessä on myös tietokantatuki muun muassa MySQL, PostgreSQL, SQLite ja Oracle tietokannoille. PHP koodia voidaan upottaa HTML:n sekaan tai käyttää sellaisenaan palvelinpuolen toiminnallisuuden toteuttamiseen. (Hudson 2006.)

Kuvassa 1. kuvataan PHP:n toimintaa web-palvelinympäristössä. Kohdassa 1. käyttäjä lähettää selaimen välityksellä pyynnön web-palvelimelle. Kohdassa 2. palvelin irrottaa pyynnöstä PHP-osan ja siirtää sen PHP-tulkille. Kohdassa 3. PHP-tulkki suorittaa PHP-

koodin ja generoi sen perusteella HTML dokumentin, jonka web-palvelin lähettää käyttäjälle kohdassa 5. (Hudson 2006.)



Kuva 1. PHP-toimintakaavio

### 2.2.1 Historia

Nykyinen PHP on PHP/FI nimisen tuotoksen jälkeläinen. Rasmus Lerdorf loi tämän ensimmäisen PHP:n ilmentymän vuonna 1994. PHP/FI oli yksinkertainen kokoelma CGI skriptejä jotka olivat kirjoitettu C-kielellä. Aluperäinen käyttötarkoitus oli seurata web-sivun kävijämäärää. Alunperin PHP oli lyhenne sanoista ”Personal Home Page Tools”. (The PHP Group 2014.)

Ajan kuluessa haluttiin lisää toiminnallisuutta ja Lerdorf uudelleenkirjoitti PHP:n, tuottaen paljon laajemman ja monipuolisemman toteutuksen. Uusi malli mahdollisti muun muassa vuorovaikutuksen tietokannan kanssa ja paljon muuta. Kesäkuussa 1995 Lerdorf julkaisi PHP Tools:n lähdekoodin, tämä mahdollisti muiden kehittäjien käyttää sitä niin kuin itse halusivat. Lisäksi kehittäjät pystyivät välittämään ohjelmointivirhe korjauksia ja yleisesti parantamaan alkuperäistä julkaisua. (The PHP Group 2014.)

Saman vuoden syyskuussa Lerdorf laajensi PHP:ta ja hylkäsi toistaiseksi nimen PHP. Uusi nimi oli FI (lyhenne sanoista ”Forms Interpreter”). Uuteen versioon sisältyi joitain perusominaisuuksia, jotka ovat tuttuja nykypäivän PHP:sta. FI:n suosio jatkoi kasvuaan CGI-työkaluna mutta ei vielä ohjelmointikielenä. (The PHP Group 2014.)

Lokakuussa 1995 Lerdorf julkaisi koodin täysin uudelleenkirjoitettuna. Uuden julkaisun nimi oli PHP ”Personal Home Page Construction Kit”. Tämä oli ensimmäinen PHP julkaisu jota aikanaan saattoi kutsua kehityneeksi skriptaus rajapinnaksi. Kieli oli

tarkoituksellisesti tehty muistuttamaan C-kieltä, jotta kehittäjien olisi helppo tottua siihen. (The PHP Group 2014.)

Huhtikuussa 1996 Ramus esitteli maailmalle PHP/FI:n. Tämä toisen sukupolven implementaatio alkoi kehittämään PHP:ta työkalujen sarjasta täysiveriseen ohjelmointikieleen. Se sisälsi tuen DBM, mSQL ja Postgres95 tietokannoille, käyttäjien määrittämille funktioille ja paljon muuta. Kesäkuussa 1996 tämä julkaisu sai versionumeron 2.0. (The PHP Group 2014.)

Kahdeksan vuoden ajan PHP:n kehittämistä jatkettiin ja se kävi läpi versiot 3 ja 4. Heinäkuussa 2004 julkaistiin PHP 5, joka on edelleenkin käytössä oleva PHP-julkaisu vaikkakin sitä on päivitetty useaan otteeseen. PHP 5 sai oman ytimensä (Zend Engine 2.0), uuden oliomallin ja tusinoittain muita ominaisuuksia. (The PHP Group 2014.)

### 2.2.2 PHP-syntaksi

PHP on suunniteltu käytettäväksi html-koodin seassa, tämän takia PHP-koodin aloitus ja lopetus on ilmoitettava `<?PHP` ja `?>` tageilla. Myös tagit `<? ja ?>` ajavat saman asian. Vaikka PHP-koodi ei olisikaan html:n seassa, vaan omissa .PHP-tiedostossaan tulee silti käyttää avaus ja lopetus tagia. (Hudson 2006.)

PHP-koodin sekaan voi kirjoittaa kommentteja, joita ei tulkita koodina. Tähän on kolme eri tapaa. Yksirivisen kommentin voi merkitä joko `//` merkillä tai vaihtoehtoisesti `#` merkillä. Monirivisen kommentin merkintään käytetään aloitusmerkkiä `/*` ja lopetusmerkkiä `*/`. (Hudson 2006.)

Muuttujien nimen edessä käytetään `$`-merkkiä. Muuttujia ei tarvitse alustaa ennen niiden käyttöä, eikä niiden tyyppiä tarvitse määrittää. Muuttujien nimet ovat merkkikorriippuvaisia, eli suurilla ja pienillä kirjaimilla on eroa. Muuttujat `$esim1` ja `$ESIM1` eivät ole sama. (Hudson 2006.)

Esimerkissä 2. on kirjoitettu PHP-skripti HTML:n väliin. Echo komennolla voidaan generoida dynaamisesti lisää sisältöä html-dokumenttiin. Tässä tapauksessa tulostetaan HTML:n taulu elementtiin sisältö. Sisemmässä for lausekkeessa tulostetaan sarakkeen

aloitus-tagi <td>, avo ja lopetus-tagi</td>. Ulommassa for lausekkeessa tulostetaan rivin aloitustagi <tr> ja rivin lopetus-tagi </tr>.

```
<table border="0">
<?PHP
    for( $rows_ = 0; $ rows_ < $ rows; $ rows_++ ){
        echo "<tr>";
        for( $cols_ = 0; $ cols _ < $ cols; $ cols _++ )
        {
            echo "<td>$value</td>";
        }
        echo "</tr>";
    }
?>
</table>
```

## ESIMERKKI 2. PHP-koodia HTML:n välissä

Funktioiden määrytykset ja kutsut hoidetaan PHP:ssa melkein kuin C-ympäristössä. Ainoa ero on, ettei funktioiden tyyppiä tarvitse määrittää vaan funktio voi palauttaa vapaasti mitä vain. (Hudson 2006.)

Esimerkissä 3. nimetään funktio `addValues`, jolle voi syöttää kaksi arvoa. Funktio laskee annetut arvot yhteen ja palauttaa tuloksen. Funktion alapuolella on esimerkki funktiokutsusta. Tällä kutsulla koodi tulostaisi luvun 8. PHP tukee myös sulkeumia, joiden avulla voidaan luoda anonyymejä funktioita, joita ei ole sidottu muuttujaan tai muuhun tunnistajaan.

PHP tukee luokkia ja abstrakteja luokkia. Luokat tukevat julkisia, yksityisiä ja suojattuja muuttujia ja funktioita. Luokkia voi myös periyttää. Luokka viittaa itseensä avainsanalla `this`. (Hudson 2006.)

```
function addValues($value1, $value2)
{
    $sum = $value1 + $value2;
    return $sum;
}
echo addValues(3, 5);
```

## ESIMERKKI 3. PHP-funktio

Esimerkissä 4. on yksinkertaisen PHP-luokan rakenne. Luokalla on kaksi yksityistä muuttujaa `$color` ja `$model`. Lisäksi luokalla on rakentaja `_construct`, jota kutsuessa voi rakentajalle syöttää arvot muuttujiin. Luokalla on myös funktio `getModel` joka palauttaa arvon olion `$model` muuttujasta. Esimerkin lopussa luodaan uusi Car-olio luokan rakentajalla ja tulostetaan olion `$model` muuttuja käyttäen `getModel()` funktiota.

```
<?PHP
class Car
{
    private string $color
    private string $model

    public function _construct($color_, $model_){
        this->$color = $color_;
        this->$model = $model_;
    }
    public function getModel(){
        return this->$model;
    }
}
$car_1 = new Car("silver", "Opel Vectra");
echo $car_1->getModel();
?>
```

**ESIMERKKI 4. PHP-luokka**

### 2.2.3 PHP-istunto

Istunnon hallinta on tekniikka, jonka avulla voidaan turvallisesti säilöä dataa yhden istunnon ajan. Istunnon ja evästeiden ero on se että istunnon dataa ei välitetä selaimelle vaan se pysyy palvelimella. Asetuksista riippuen palvelin säilyttää istunnon datan muistissa tai tiedostossa. (Jyväskylän yliopisto 2014.)

Silloin, kun istunto käynnistetään, sillä luodaan yksilöllinen istuntotunniste. Istuntoon voidaan sen jälkeen asettaa haluttu määrä ns. istuntomuuttujia. Istuntotietoon päästään

möyhemmin käsiksi istuntotunnisteteiden avulla. Tämän vuoksi istuntotunniste on jollain tavalla välitettävä aina selaimelle. (Jyväskylän yliopisto 2014.)

Istuntoon viitataan PHP:ssa `$_SESSION`-muuttujan avulla. Siihen viitataan ns. istuntomuuttujan nimen avulla. Uuden istuntomuuttujan alustaminen tapahtuu näin: `$_SESSION['nimi'] = "jotain";`. (Jyväskylän yliopisto 2014.)

Istuntomuuttujaan voidaan viitata heti sen asettamisen jälkeen. Tämä on ero evästeisiin, jotka ovat käytettävissä vasta seuraavalla kerralla, kun sivu ladataan. Istuntomuuttujista on kuitenkin huomattava, että ennen niiden käyttöä on jokaisella sivulla kutsuttava `session_start()`-funktioita. Tämä funktio tutkii ensin löytääkö se jo käynnissä olevan istunnon. Jos näin tapahtuu se alustaa automaattisesti `$_SESSION` taulukkoon käytössä olevat istuntomuuttujat. Jos istuntoa ei löydy luodaan uusi. (Jyväskylän yliopisto 2014.)

Tietoturvasyistä tulisi istunto tuhota, kun sitä ei enää tarvita. Muuten istuntodata jää palvelimelle muistiin ja voi joutua väärin käsiin mahdollisen tietomurron sattuessa. Yksittäisen istuntomuuttujan voi tuhota `unset($_SESSION['muuttuja']);`-funktioilla. Kaikki istuntomuuttujat saa tuhottua `session_unset();`-funktioilla. (Jyväskylän yliopisto 2014.)

#### 2.2.4 URL-parametrit

URL-parametreillä tarkoitetaan muuttujia, jotka välitetään sivulta toiselle selaimen URL:ssa. Muuttujat ja URL-osoite erotetaan toisistaan `"?"`-merkillä. Useampia muuttujia välittäessä erotetaan ne toisistaan `"&"`-merkillä. Jos muuttujien arvoissa on välilyöntejä, ne korvataan `"+"`-merkeillä. Esimerkki 5. kuvaa URL:aa, johon sisältyy parametreja. (The World Wide Web Consortium 2014.)

`http://nettisivu.fi/views/view1?nimi=Pasi&iika=12`

ESIMERKKI 5. URL jossa välitetään parametreja

Parametrit tallennetaan URL:iin HTML:n hyperlinkin avulla käyttäen aiemmin mainittuja sääntöjä. Parametrien tallentamiseen saadaan dynaamisuutta helposti PHP:n avulla. (The World Wide Web Consortium 2014.)

Parametreihin pääsee käsiksi PHP-koodissa superglobaalin `$_GET`-muuttujan avulla. Parametrejä käsitellään kun ne olisi tallennettu taulukkoon. Esimerkiksi URL:iin tallennettuun ”nimi” muuttujan arvo luetaan `$_GET["nimi"];` komennolla. `$_GET`-muuttujan tallennettuja arvoja ei kuitenkaan voi käsitellä niinkuin normaaleja PHP-muuttujia vaan se on vain lukemista varten. Jos arvoa halutaan muuttaa pitää se tallentaa muuttujan, jonka jälkeen sitä voi käsitellä normaalina muuttujana. (The World Wide Web Consortium 2014.)

### 2.2.5 Evästeet

Evästeet ovat tekstimuotoista dataa, jota palvelin voi lähettää selaimelle. Evästeelle voidaan määrittää haluttu elinikä, jonka ajan selain säilyttää evästeeseen tallennetun datan. Tyypillisesti selain säilyttää evästeen sen ajan kuin istunto on voimassa. Jos evästeen elinikä on määritelty pidemmäksi, tallentaa selain sen tiedostoon. Seuraavalla kerralla, kun selain kutsuu sivua se lähettää evästeen palvelimelle, jolloin palvelimella toimiva www-sovellus voi hyödyntää edellisellä kerralla talletettua dataa. (Jyväskylän yliopisto 2014.)

Evästeessä voidaan säilyttää esimerkiksi ostoskorin tiedot verkkokaupassa, jolloin käyttäjä voi jättää ostoksensa kesken ja jatkaa niitä sujuvasti seuraavalla kerralla. Evästeeseen ei tule tallentaa mitään arkaluontoista tietoa, sillä ne ovat kenen tahansa luettavissa olevaa tekstimuotoista tietoa. Evästeeseen ei myöskään kannata tallentaa mitään elintärkeää sillä ne on helposti poistettavissa. (Jyväskylän yliopisto 2014.)

PHP-koodissa evästeen asettaminen toteutetaan `setcookie()`-funktiolla. Funktiota käytettäessä on muistettava, että sitä täytyy kutsua ennen XHTML-koodin alkamista PHP-sivulla. Eli käytännössä kutsu täytyy olla ennen XML-versiomäärittelyn tulostamista. (Jyväskylän yliopisto 2014.)

Esimerkissä 6. setcookie funktiolle annetaan ensimmäisenä parametrina muuttujan nimi, joka on tässä tapauksessa **nimi**. Toisena parametrina funktiolle syötetään muuttujan arvo, joka tässä tapauksessa on **Pasi**.

Funktion kolmas parametri on aika, jonka eväste säilyy. Aika annetaan ”Unix timestamp”-muodossa. ”Unix timestamp” on sekunteina ilmoitettu aika, joka on kulunut vuoden 1970 alusta. PHP-funktio `time()` palauttaa tämänhetkisen ajan tuossa muodossa, kun siihen lisätään sekunteina se aika kuinka kauan evästeen halutaan pysyvän tallessa saadaan parasta ennen aika kyseiselle evästeelle. Tässä tapauksessa eväste säilyy 30 vuorokautta. (Jyväskylän yliopisto 2014.)

Funktion neljäs parametri on evästeen yleisen asettamisen yhteydessä käsitelty ”path”-asetus. Viides parametri on samassa yhteydessä käsitelty ”domain”-asetus. (Jyväskylän yliopisto 2014.)

Kuudes ja viimeinen parametri on ”secure”-asetus. Sen arvo määrittää voidaanko eväste siirtää suojaamattoman yhteyden yli vai vaatiiko se https-protokollan käyttöä. Arvo 0 ei vaadi suojattua yhteyttä, arvo 1 vaatii. (Jyväskylän yliopisto 2014.)

```
setcookie( "nimi", "Pasi", time()+60*60*24*30, "/",
".tamk.fi", "1" );
```

ESIMERKKI 6. PHP-setcookie funktion käyttö

## 2.3 JavaScript

JavaScript on olennainen osa lähes jokaista web-sivustoa. Se on pääasiallinen tapa muokata dynaamisesti käyttäjälle näkyviä DOM-elementtejä ja se mahdollistaa myös AJAXin avulla datan dynaamisen noutamisen serveriltä. Koska AJAXilla voidaan käsitellä HTTP-pyyntöjä, voidaan dynaaminen data päivittää lataamatta koko sivua uudelleen. Javascriptiä käytetään nykyisin monesti kirjastojen kuten jQueryn kautta.

### 2.3.1 Kirjastojen käyttö verrattuna natiiviin JavaScriptiin



Monesti JavaScriptin yksi ongelmista on toiminnallisuuden erottaminen muusta koodista. Esimerkiksi DOM-elementtien muokkaus tapahtuu viittaamalla elementtiin suoraan JavaScript-koodista, jolloin elementistä itsestään ei näe viittausta toiminnallisuuteen. Tämän haittapuolen pystyy osin kiertämään esimerkiksi AngularJS-kirjastolla, jossa viittaukset funktionaalisuuteen usein säilytetään elementeissä itsessään. Kirjastot sisältävät luonnollisesti valmiita funktioita jotka on varustettu oletusarvoilla. Tämä vähentää tarvittavan koodin määrää ja samoin selkeyttää rakennetta. Käytön edut näkee selkeästi jos verrataan esimerkiksi tyypillistä AJAX-pyyntöä JavaScriptillä(Esimerkki 7.), ja samaa toimintoa käytettynä jQueryn kautta(Esimerkki 8.). (Sitepoint 2014.)

```
var xhr = new XMLHttpRequest();
xhr.open('get', 'send-ajax-data.PHP');

xhr.onreadystatechange = function() {
    if(xhr.readyState === 4) {
        if(xhr.status === 200) {
            alert(xhr.responseText);        } else {
            alert('Error: '+xhr.status);
        }
    }
}
```

```
xhr.send(null);
```

ESIMERKKI 7. AJAX-pyyntö JavaScriptillä

```
$.get('send-ajax-data.PHP', function(data) {
    alert(data);
});
```

ESIMERKKI 8. AJAX-pyyntö jQuery-kirjastoa käyttäen

Luonnollisesti tällä toimintatavalla on myös miinuspuolia. jQuery ei ole yhtä joustava tilanteissa, joissa halutaan optomoida koodia. jQuery myös menettää yksinkertaisuutensa etuja pidemmälle yksilöidyissä kutsuissa. jQueryn käyttö vaatii myös lisäfunktioiden käytön opettelua, eikä koodi ole aina itseselitteistä. Esimerkiksi jos halutaan lisätä lennosta erilainen vastaus koodille 404, täytyy kaivaa referenssilista esille. JavaScriptin tapauksessa status on määritelty joka tapauksessa ja oikean ratkaisun voi päätellä täysin tuntematta syntaksia. (Sitepoint 2014.)

### 3 TIETOKANTA

Tietokanta on tietotekniikassa käytetty termi tietovarastolle. Tietokannassa data esitetään taulukkorakenteella. Data on tauluissa riveillä ja sarakkeissa, jokaisella sarakkeella on jokin tietotyyppi joka kertoo millaista dataa sarakkeeseen voi tallentaa. Tarvittavat tiedot pyritään jakamaan taulukkoihin siten, että yksi tieto tallennetaan vain yhteen paikkaan. (Wikipedia: Tietokanta 2014.)

#### 3.1 Relaatiotietokanta

Relaatiotietokannoissa taulujen välille luodaan yhteyksiä eli relaatioita. Yhteyksien avulla tiedetään miten eri taulut liittyvät toisiinsa. Taulujen välisiä suhteita voidaan kuvata äiti-lapsi-termillä. Tarkoittaen että äidillä voi olla useita lapsia mutta lapsella voi olla vain yksi äiti, tätä kutsutaan one-to-many suhteeksi(1 : n). Taulujen tiedot yhdistetään toisiinsa toisen taulun avaimella. Yleensä käyttäen ID:tä. Toisessa taulussa yhteyden luovaa saraketta kutsutaan vierasavaimeksi. Kuvassa 2. on havainnollistettu kahden taulun välistä suhdetta. (Wikipedia: Tietokanta 2014.)

työntekijä			N : 1	osasto	
työntekijä_id	nimi	osasto		osasto_id	nimi
1001	Matti	1		1	osto
1002	Maija	1		2	myynti
1003	Teppo	2		3	huolto

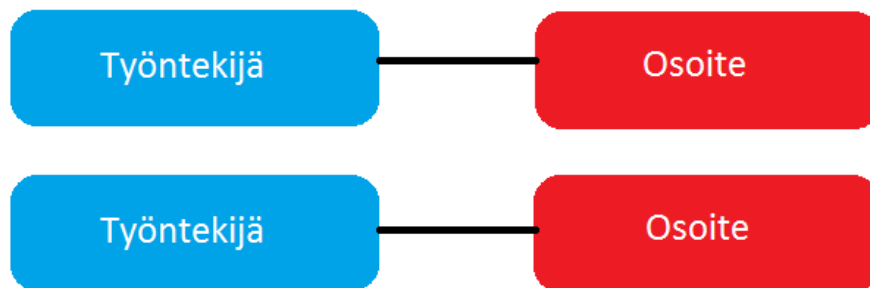
Kuva 2. Esimerkki kahdesta taulusta ja niiden yhteydestä.

Yhdistettyjen taulujen tietojen käsittelyä hallitaan viite-eheyksillä. Viite-eheydet määrittävät mitä yhdistetyille tiedoille tapahtuu, jos taulujen tiedot vaihtuvat tai ne poistetaan. Viite-eheyksillä voidaan siis määrittää esimerkiksi poistuvatko lapsitiedot jos äititieto poistetaan. (Wikipedia: Tietokanta 2014.)

### 3.1.1 Yhteydet

#### Yksi-yhteen-yhteys

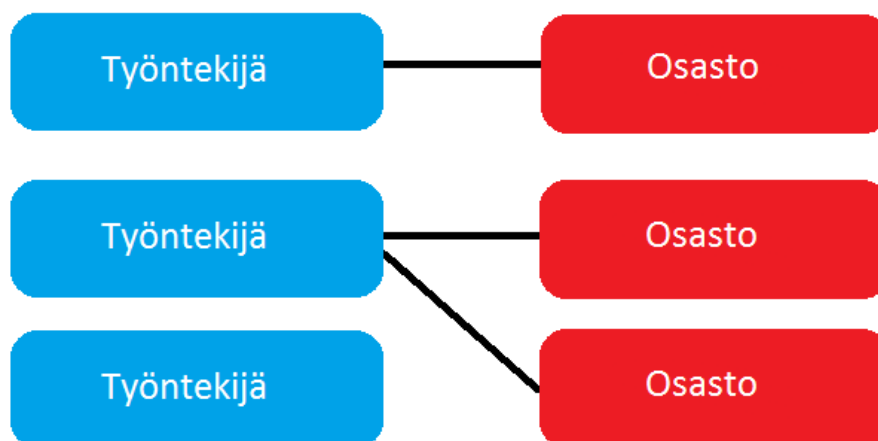
One to One(1:1) eli yksi-yhteen-yhteys on harvoin käytetty suhde, jossa ”äidillä” voi olla vain yksi ”lapsi”. Tätä yhteyttä käytetään harvoin siksi, että tämällyylyset tiedot tallennetaan yleensä yhteen tauluun. 1:1-yhteyttä voidaan kuitenkin käyttää esimerkiksi, jos halutaan tallentaa vaikka työntekijän osoite erilliseen tauluun. Tällöin yhdellä työntekijällä voi olla vain yksi osoite ja tietty osoite voi kuulua vain yhdelle työntekijälle. Tämän yhteyden rakennetta on havainnollistettu kuvassa 3. (Tuts+: SQL for Beginners part 3 2014.)



Kuva 3. 1:1-yhteys

#### Yksi-moneen-yhteys

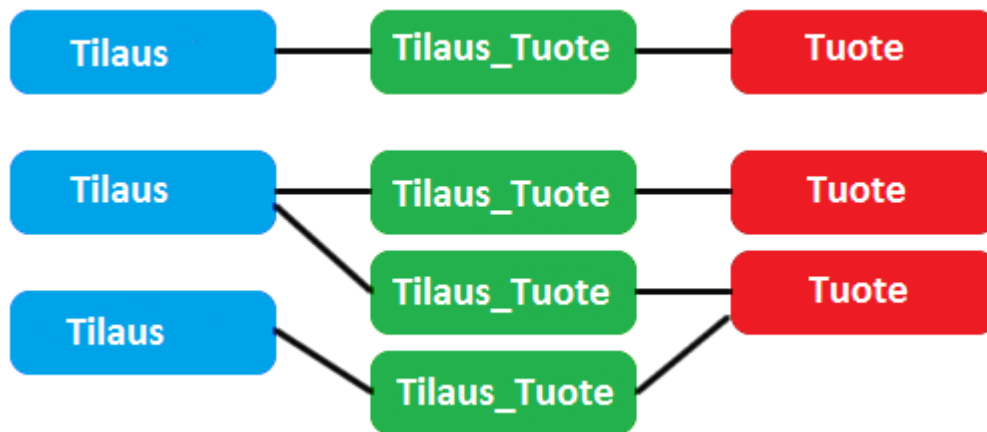
One to Many(1:n) eli yksi-moneen-yhteys on yleisimmin käytetty yhteys, jossa ”äidillä” voi olla monta ”lasta” ja lapsella yksi äiti. Esimerkkinä yrityksessä monta työntekijää voi työskennellä samassa osastossa mutta työntekijä voi työskennellä vain yhdessä osastossa. Tämän yhteyden rakennetta on havainnollistettu Kuvassa 4. (Tuts+: SQL for Beginners part 3 2014.)



Kuva 4. 1:n-yhteys

### Moni-moneen-yhteys

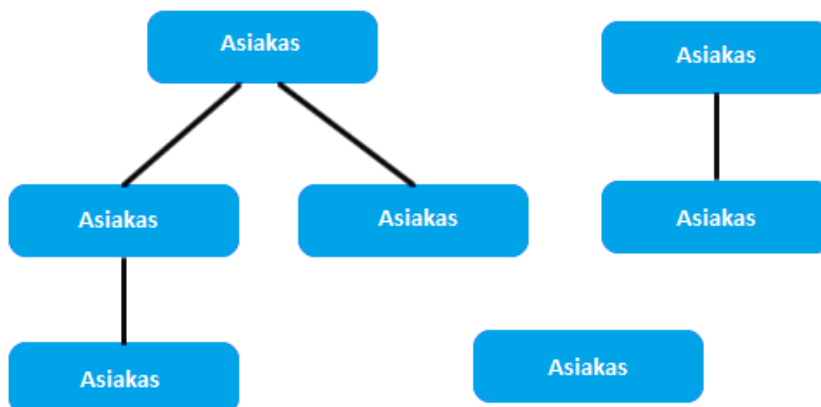
Many to Many(n:n) eli moni-moneen-yhteyttä tarvitaan silloin kuin “lapsella” voikin olla monta “äitiä”. Esimerkkinä yrityksen tilaustietokannassa on monta tilausta, jotka sisältävät saman tuotteen. N:n-suhteen toteutuksessa tarvitsee taulujen väliin lisätä liitostaulu. Liitostaulu kytkee kahden tai useamman taulun tietoja yhteen. Kuvan 5. tapauksessa liitostaulussa olisi sarakkeet tilaus\_id sekä tuote\_id. (Tuts+: SQL for Beginners part 3 2014.)



Kuva 5. N:n-yhteys

### Itseensä-viittaava-yhteys

Self Referencing eli itseensä-viittaava-yhteys tarvitaan silloin kun halutaan luoda suhde samassa taulussa sijaitsevien tietueiden välille. Esimerkkinä järjestelmä jossa asiakas voi jotain etua vastaan kutsua kavereitaan asiakkaaksi. Tällöin uuden kutsutun asiakkaan tietueeseen tallennetaan kutsujan id. Tämän yhteyden rakenne on havainnollistettu kuvassa 6. (Tuts+: SQL for Beginners part 3 2014.)



Kuva 6. Itseensä-viittaava-yhteys

## 3.2 MySQL

MySQL on maailman toiseksi laajimmin käytetty avoimen lähdekoodin relaatiotietokannan hallintajärjestelmä. Sillä on useita miljoonia käyttäjiä, mukaanlukien yksittäisiä henkilöitä ja suuria yrityksiä kuten Facebook ja Youtube. (Wikipedia: MySQL 2014.)

MySQL:n kehitti suomalainen Michael ”Monty” Wildenius yhdessä ruotsalaisen David Axmarkin kanssa vuonna 1995. MySQL:n ensimmäinen versio julkaistiin vuonna 1996. Uusin ja suositeltava versio on 5.0. Oracle Corporation osti MySQL:n omistavan Sun Microsystemsin huhtikuussa 2009 ja näin omistus siirtyi Oraclelle. (Wikipedia: MySQL 2014.)

MySQL on GNU-lisenssin alainen ohjelmisto. Tämä tarkoittaa sitä että kenellä vain on oikeudet käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmaa ja sen lähdekoodia. (Wikipedia: MySQL 2014.)

MySQL on saatavilla useille eri alustoille kuten Linuxille, Microsoft Windowsille, OS X:lle, Solarikselle, Symbianille ja monelle muulle. (Wikipedia: MySQL 2014.)

MySQL on kirjoitettu C:llä ja C++:lla. Sen SQL-jäsentäjä on kirjoitettu yacc:lla, mutta se käyttää kotitekoista lexicaalista analysointia. Monet ohjelmointikielet sisältävät kirjastoja MySQL-tietokannan käsittelyyn. Näihin kuuluvat mm. MySQL Connector/Net, jota käytetään Microsoft Visual Studion kanssa esim. C# ja VB ohjelmointikielillä. (Wikipedia: MySQL 2014.)

## 3.3 SQL

SQL eli Structured Query Language on standartoitu kieli, jota käytetään relaatiotietokantojen kyselyissä, päivityksissä ja hallinnassa. SQL:n avulla voidaan hakea, lajitella ja suodattaa tietokannasta haettavia tietoja. SQL:ää tukevat mm. MS Access, SQL Server, MySQL ja SQLite tietokannan hallintajärjestelmät. SQL-kyselyjä voidaan kirjoittaa tietokantaohjelmiston käyttöliittymässä, tai niitä voidaan generoida ohjelmallisesti osana jotain sovellusta. (Koulutuskeskus Salpaus 2014.)

### 3.3.1 Syntaksi

Tyypillisellä SQL-kyselyllä on esimerkin 9. kaltainen rakenne. Jossa  $A_i$  kuvaa haettavia attribuutteja,  $r_i$  kuvaa relaatioita eli mistä haetaan,  $\rho$  kuvaa predikaattia eli minkä perusteella tiedot valitaan ja  $\sigma$  kuvaa valinta komentoa. Tälläisen kyselyn matemaattinen malli on seuraavanlainen:  $\Pi A_1, A_2, \dots, A_n (\sigma \rho (r_1 r_2 \dots r_m))$ . (New York University 2014.)

**select**  $A_1, A_2, \dots, A_n$

**from**  $r_1, r_2, \dots, r_m$

**where**  $\rho$

ESIMERKKI 9. SQL-kyselyn rakenne

#### Taulun rakentaminen

Uuden taulun rakentamiseen käytetään CREATE TABLE-lausetta. Lause määrittää uuden taulun nimen, kentät ja kenttärajoitukset. Esimerkissä 10. luodaan taulu "tuote". Taululle luodaan sarakkeet "tuote\_id", "nimi", "hinta" ja "luotu". Jokaiselle sarakkeelle määritetään tietotyyppi, mahdolliset rajoitukset ja avaimet. Esimerkissä "tuote\_id" määritetään uniikiksi, ettei kahdella tietueella voi olla samaa id:tä. Lisäksi "tuote\_id":lle määritetään AUTO\_INCREMENT ominaisuus, joka automaattisesti kasvattaa "tuote\_id":n arvoa uutta tietuetta tallentaessa. NOT NULL rajoutuksella estetään sarakkeen tyhjäksi jättäminen, sarakkeen tyhjäksi jättämisestä seuraisi virheilmoitus. Luotu sarakkeeseen tallennetaan sen hetkinen aika, ellei käyttäjä muuta määritä. Lisäksi taululle määritetään vierasavain, joka osoittaa varasto taulun "varasto\_id" sarakkeeseen. (Koulutuskeskus Salpaus 2014.)

```
CREATE TABLE tuote (
  tuote_id INTEGER UNIQUE AUTO_INCREMENT PRIMARY KEY NOT NULL,
  nimi VARCHAR(50),
  hinta FLOAT,
  luotu TIMESTAMP DEFAULT NOW(),
  FOREIGN KEY (varasto_id) REFERENCES varasto(varasto_id)
);
```

ESIMERKKI 10. Taulun rakentaminen

### Tietojen lisääminen

Uuden tiedon lisääminen tauluun suoritetaan INSERT INTO-lauseella. INSERT INTO-komennon jälkeen määritetään mihin tauluun tiedot lisätään. Tämän jälkeen määritetään mitä tietoa lisätään. Esimerkissä 11. lisätään aiemmin luotuun ”tuote” tauluun uusi tietue. Sarakkeille nimi ja hinta on määritetty tallennettavat arvot. Sarakkeet ”tuote\_id” ja ”luotu” saavat arvot automaattisesti tietokantaohjelmistolta edellisen esimerkin määrittämien perusteella. Tauluun voi lisätä tietoa myös kirjoittamatta sarakkeiden nimiä INSERT komentoon. Tällöin values kohtaan kirjoitetaan suoraan tallennettavat arvot pilkuin erotettuna. Näin tehdessä pitää olla tarkkana, että arvot kirjoitetaan samassa järjestyksessä kuin ne ovat taulussa. (Koulutuskeskus Salpaus 2014.)

```
INSERT INTO tuote
  (nimi, hinta)
VALUES
  ('Tuote1', 12,95);
```

ESIMERKKI 11. Tiedon lisääminen tauluun

### Tietojen haku

Tietokannasta haettaessa dataa käytetään SELECT-lausetta. SELECT-lauseen avulla voidaan etsiä määritetyn taulukon tai taulukoiden valitut sarakkeet, valita ehdot täyttävät rivit ja lajitella tai ryhmitellä tulosrivit haluttuun järjestykseen. (Koulutuskeskus Salpaus 2014.)

<b>SELECT ...</b>	Määrittää haettavat sarakkeet.
<b>FROM ...</b>	Määrittää mistä tieto haetaan. Tietoa voi hakea yksittäisestä tai useammasta taulusta tai näkymästä.
<b>WHERE ...</b>	Where ehdon avulla voidaan rajata kyselyn palauttamaa tulosjoukkoa, asettamalla palautettavalle datalle kelpoisuusrajoituksia. Hyödyllisiä loogisia operaattoreita where lauseessa ovat <b>AND</b> , <b>OR</b> , <b>NOT</b> ja <b>BETWEEN</b> .
<b>GROUP BY ...</b>	Ryhmittelee tietueet sarakkeen mukaan, siten ettei palautuksessa ole identtisiä sarakkeita.
<b>HAVING ...</b>	Having lauseen avulla voidaan rajata tulosjoukkoa edelleen käyttäen SQL:n agregaatti funktioita.
<b>ORDER BY ...</b>	Määrittää minkä sarakkeen perusteella tietueet järjestetään.

TAULUKKO 1. Select lauseen muodostamisessa käytettävät osat (Koulutuskeskus Salpaus 2014.)

Esimerkissä 12. on yksinkertainen SQL-kysely jossa valitaan tuote taulusta kaikki ne tietueet, joissa hinta on alle 50. Tämän jälkeen ne järjestetään hinnan mukaan ja tulosjoukko palautetaan.

```
SELECT *
FROM tuote
WHERE hinta < 50.00
ORDER BY hinta;
```

ESIMERKKI 12. Yksinkertainen SELECT-lause

Usein tietokantakyselyissä tarvitsee yhdistellä useamman taulun tietoja. Esimerkkinä tilanne, jossa kahden taulun välinen suhde on N:N ja niiden välissä on liitostaulu. Esimerkissä 13. kahden taulun sarakkeet yhdistetään toisiinsa, niin että tuote\_id on niillä yhteinen. Sitten etsitään rivit, joiden tilaus\_id on 1001 ja lasketaan niiden hinta-sarakkeiden arvot yhteen.

```
SELECT SUM(hinta)
FROM tilaus_sisaltaa_tuotteen m INNER JOIN tuote n
ON m.tuote_id = n.tuote_id
WHERE tilaus_id = 1001
```

ESIMERKKI 13. JOIN-ominaisuuden käyttö SQL-kyselyssä



## Tietojen muuttaminen

Tietojen muuttamisella tarkoitetaan jo taulussa olevan datan muuttamista. Tietojen muuttaminen tapahtuu UPDATE-lauseella. Jonka rakenne on kuvattu esimerkissä 14.

```
UPDATE taulun_nimi
SET sarakkeen_nimi = arvo [, toinen_sarake = arvo ... ]
[WHERE hakuehdot]
```

ESIMERKKI 14. UPDATE-lauseen rakenne

Jos hakuehdot jätetään määrittämättä niin muutos tehdään taulun jokaiselle riville. UPDATE-lauseessa voidaan käyttää myös JOIN-liitoksia. (Koulutuskeskus Salpaus 2014.)

### 3.4 Kykylaakson tietokanta

Kykylaakson tietokanta ja sen käsittely ovat järjestelmän suurin ongelmakohta. Tietokannan viite-eheys on lähes olematonta ja tämän takia tietokanta on täynnä monistunutta ja turhaa dataa. Osassa tauluissa on lähes sata saraketta ja tästä syystä lähes identtisiä rivejä on satoja. Tältä oltaisiin voitu välttyä, jos tietokannan suunnittelussa oltaisiin käytetty tehokkaammin taulujen välisiä suhteita. Jos olisi käytetty moni-moneen ja yksi moneen suhteita oikein, olisi saatu sama data tallennettua käyttäen vähemmän tilaa tietokannasta. Kykylaakson SQL-kyselyt ovat sekavia ja epäloogisia verrattuna nykyaikaisiin toteutustapoihin ja tämän takia vianselvityksessä täytyy olla tarkkana.

Tietokantakyselyitä on lukemattomia määriä PHP-koodin seassa kaikkialla eikä niitä ole kommentoitu. Usein paras keino tarkastella koodin toimintaa on luoda omaa testidataa ja tarkastella miten kyseinen koodi sitä käsittelee.

Nykypäivänä useissa järjestelmissä tietokanta sidotaan malliin, jolloin voidaan käyttää NoSQL-ratkaisua. Tällaisissa järjestelmissä eheys, relaatiot, rakenne ja tiedon tyyppitys hallitaan käytettävässä sovelluksessa tietokannan sijaan. Eli tietokannasta luodaan virtuaalinen oliokanta, joka liitetään rajapinnalla varsinaiseen tietokantaan.

## 4 PALVELINYMPÄRISTÖ

Palvelin on tietokone, joka tarjoaa siinä ajettavien palvelinohjelmistojen välityksellä erilaisia palveluja muille ohjelmille. Asiakkaana toimivat ohjelmat voivat sijaita joko samalla koneella tai eri koneilla. Eri palvelimilla on eri rooleja, kuten www-palvelin, nimipalvelin, sähköpostipalvelin jne. . (Linux.fi – wiki: Palvelin 2014.)

Kykylaakso on hajautettu kolmelle eri palvelimelle, www-palvelimelle, tietokantapalvelimelle ja sähköpostipalvelimelle. Järjestelmän kehittäjän kannalta on www-palvelin oleellisin, sillä siellä sijaitsee järjestelmän käyttäjälle näkyvä osuus. Kykylaaksolla on erikseen testipalvelin ja tuotantopalvelin. Näin kehitys on turvallisempaa ja saadaan eliminoidua turhat käyttökatkokset.

Fyysisesti kaikki kykylaakson palvelimet ovat samassa laitteessa, mutta ne ovat virtualisoitu omiksi palvelimikseen. Kykylaakson tietokanta- ja sähköpostipalvelimet ovat myös muiden projektien käytössä.

### 4.1 Debian GNU/Linux

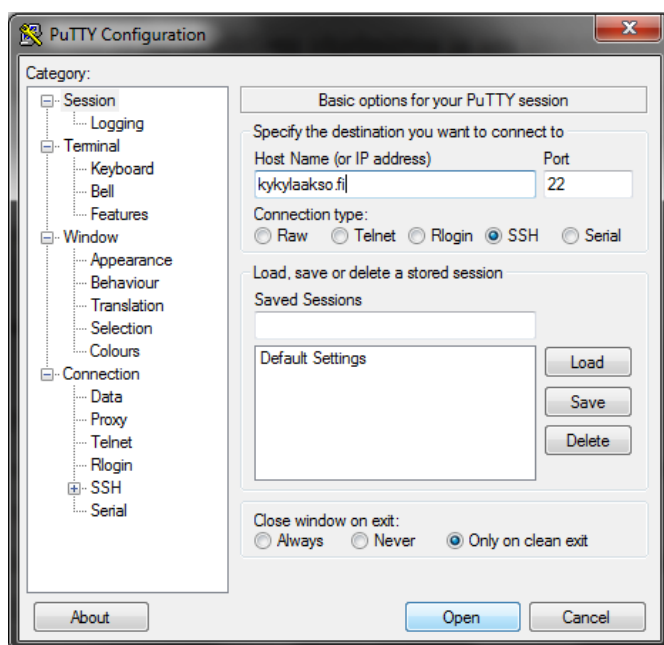
Kykylaakso palvelimen käyttöjärjestelmänä toimii Debian GNU/Linux. Debian käyttöjärjestelmää saa usealla eri ytimellä, mutta tässä versiossa on Linux ydin. Debian GNU/Linux on yksi suosituimmista Linux jakeluista ja se on ollut pohjana 144:lle muullekin Debian jakelulle. (Wikipedia: Debian 2014.)

Debian on GPL-lisenssin alainen ohjelmisto, eli kuka tahansa saa käyttää, kopioida, muuttaa ja jakaa edelleen ohjelmistoa ja sen lähdekoodia. Lisäksi lisenssi takaa, että nämä vapaudet säilyvät myös GPL-koodiin pohjautuvissa muunnelluissa teoksissa (derivative work). Mikäli GPL-ohjelmaa tai sen muunnelmaa levitetään edelleen, lähdekoodi on julkaistava samalla lisenssillä eikä ohjelman käytölle tai levitykselle saa asettaa lisärajoituksia. (Wikipedia: Debian 2014.)

### 4.1.1 Palvelimeen yhdistäminen

Palvelimelle on asennettu SSH-palvelinohjelmisto, joka mahdollistaa turvallisen ja salatun verkkoyhteyden palvelimen komentoriiviin yhdistäessä. SSH-yhteyden voi avata Windows-käyttöjärjestelmästä SSH-asiakasohjelmalla kuten Putty. Linux käyttöjärjestelmällä yhdistäessä tarvitaan myös SSH-asiakasohjelmisto mutta useissa Linux jakeluissa sellainen on asennettuna valmiiksi.

Puttyä käyttäessä kirjoitetaan vain palvelimen osoite ”Host Name” kenttään ja painetaan ”Open” painiketta(Kuva 7.). Kun yhteys on avattu kysyy palvelin käyttäjätunnusta ja salasanaa. Tämän jälkeen eteesi avautuu palvelimen komentoriivi.



Kuva 7. SSH-yhteyden avaaminen Putty:lla.

Linux käyttöjärjestelmällä yhdistäessä kirjoitetaan komentoriiviin komento ”ssh käyttäjätunnus@kykylaakso.fi”.

Kykylaakson testipalvelimelle ei voi muodostaa yhteyttä TAMK:n verkon ulkopuolelta. Testipalvelimeen saa kuitenkin yhteyden, kun ottaa ensin SSH-yhteyden TAMK:n verkkoon osoitteella ”tpu.fi” käyttäen omia TAMK:n verkon tunnuksia. Tämän jälkeen avataan edelleen yhteys kykylaakson testipalvelimelle.

## 4.1.2 Hyödyllisiä komentoja

### Navigoiminen

Linuxympäristössä navigoiminen on hieman vaivalloista sillä käyttöliittymä on täysin merkistöperäinen ja navigoiminen täytyy suorittaa kirjoittamalla komentoja komentoriville. Taulukossa 2. on lista komentoista, joilla voi navigoida tiedostorakenteessa. (Linuxcommand.org 2014.)

<b>pwd</b>	Tulostaa tiedostopolun, jossa sijaitset kyseisellä hetkellä.
<b>cd</b>	Avaa käyttäjän kotihakemiston.
<b>cd /</b>	Avaa järjestelmän juurikansion. Voidaan myös jatkaa tyyliin ”cd /kansio/tiedosto”.
<b>cd ..</b>	Siirry tiedostorakenteessa yksi kansi taaksepäin. Voidaan käyttää myös tyyliin ”cd ../kansio”, jolloin mennään yksi kansio taaksepäin ja avataan sieltä tiedosto tai kansio.
<b>cd -</b>	Siirry kansioon jossa olit viimeksi.
<b>ls</b>	Tulostaa listan tiedostoista ja kansioista, jotka sijaitsevat avoimena olevassa kansiossa. Lisäasetuksella ”-l” tulostetaan tarkempaa tietoa tiedostoista, kuten kenelle se kuuluu, koska se on luotu ja millaiset käyttöoikeudet sille on asetettu.

TAULUKKO 2. Navigoimisessa käytettävät komennot

### Grep

Grep on Unix-käyttöjärjestelmän komentoriviohjelma, joka etsii säännöllisillä lausekkeilla määritellyjä merkkijonoja tiedostoista. Grep-komennon perus rakenne on ”grep [asetukset] merkkijono [tiedosto...]”. Esimerkiksi haettaessa merkkijonoa useasta kansioista voidaan käyttää komentoa ”grep ‘merkkijono’ tiedosto1 tiedosto2”. Vaihtoehtoisesti voidaan tiedoston nimen sijaan käyttää \*-merkkiä, jolloin merkkijonoa haetaan kansion jokaisesta tiedostosta. Jos \*-merkin perään kirjoittaa tiedosto päätteen niin haetaan merkkijonoa vain sen tyyppisistä tiedostoista. Jos halutaan hakea asempaa merkkijonoa kerrallaan voidaan ne erottaa toisistaan |-merkillä. Haun onnistuessa grep tulostaa oletuksena sen rivin, jossa merkkijonot täsmäsivät. (Wikipedia: Grep 2014.; Computer Hope 2014.)

Grep-hakuun voidaan määrittää myös useita eri asetuksia, joilla voidaan optimoida hakua. Taulukkoon 3. on listattu näitä asetuksia

<b>r</b>	Haetaan myös nykyisen hakemiston alihakemistoista.
<b>a</b>	Luetaan läpi myös binääritiedostot niinkuin ne olisivat tekstitiedostoja.
<b>l</b>	Ei tulosta löytyneitä rivejä, vaan niiden tiedostojen nimet joista osuma löytyi.
<b>c</b>	Normaalin tulostuksen sijaan tulostetaan kuinka monta osumaa kyseinen haku tuotti.
<b>i</b>	Pidä isoja ja pieniä kirjaimia samana.
<b>n</b>	Tulostaa rivinumerot ennen osumarivejä.
<b>w</b>	Hae vain rivit, joissa hahmo esiintyy yhtenä sanana, ei sanan osana.

Taulukko 3. Grep-haun asetukset (Computer Hope 2014.)

### **Tiedostojen käyttöoikeudet**

Linuxissa jokaisella tiedostolla, hakemistolla, laitetiedostolla ym. on tietyt oikeudet, jotka kertovat ketkä tiedostoa saavat lukea, kirjoittaa ja suorittaa. Nämä oikeudet parantavat huomattavasti tietoturvaa. Jokaisella tiedostolla on kolme erillistä oikeutta: kirjoitusoikeus, lukuoikeus ja suoritusoikeus. Voidaan esimerkiksi määrätä, että tiedostoa saavat lukea kaikki käyttäjät mutta vain root kirjoittaa ja suorittaminen ohjelmatiedostona ei ole sallittua kenellekään. Tällaisia tiedostoja ovat esimerkiksi lähes kaikki /etc-hakemiston asetustiedostot. (Linux.org 2014.)

Tiedoston oikeudet voidaan merkitä joko numerosarjalla (esimerkiksi 660) tai kirjainyhdistelmällä (esimerkiksi -rw-r-----). Taulukossa 4. on selitetty numeroiden ja kirjainyhdistelmien merkitykset. (Linux.org 2014.)

Numeromuoto	Kirjainyhdistelmä	Oikeudet
0	---	ei oikeuksia
1	--x	suoritus
2	-w-	kirjoitus
3	-wx	kirjoitus ja suoritus (1+2)
4	r--	luku
5	r-x	luku ja suoritus (1+4)
6	rw-	luku ja kirjoitus (2+4)
7	rwX	luku, kirjoitus ja suoritus (1+2+4)

Taulukko 4. Käyttöoikeuksien numeroiden merkitykset.

Numeroyhdistelmässä ensimmäinen numero tarkoittaa tiedoston omistajan oikeuksia, toinen tiedoston ryhmän oikeuksia ja kolmas muiden oikeuksia. Jokaisella tiedostolla on omistaja (joka on oletuksena tiedoston luoja) ja se kuuluu johonkin käyttäjäryhmään. (Linux.org 2014.)

Tiedoston tai hakemiston oikeudet asetetaan komentorivillä käyttäen **chmod**-ohjelmaa. Tiedoston omistaja ja pääkäyttäjä voivat muuttaa sen oikeuksia esimerkiksi komennolla: ”chmod 760 tiedosto”. Jos komennon kanssa käytetään valitsinta -R ja tiedosto on hakemisto, muutetaan myös jokaisen alihakemiston ja niiden alihakemistojen tiedostojen oikeudet. (Linux.org 2014.)

Tiedostojen tai kansioiden käyttöoikeudet saa tulostettua käyttämällä ”stat tiedosto”-komentoa komentorivillä. Tämä komento tulostaa käyttöoikeudet numerosarjana ja kirjainyhdistelmänä. Lisäksi stat komento tulostaa muutakin hyödyllistä tietoa tiedostosta. (Linux.org 2014.)

### Tiedostojen kopioiminen

Välillä tulee vastaan tilanne kun tarvitsee kopioida tiedostoja paikasta toiseen. Esimerkiksi silloin kun siirretään testiympäristössä muokattu tiedosto tuotantopalvelimelle.

Tiedostojen kopioinnissa käytetään komentoa ”cp”. Tämän komennon syntaksi on kuvattu esimerkissä 15. (Computer Hope 2014.)

```
cp [OPTION]... [-T] SOURCE DEST
cp [OPTION]... SOURCE... DIRECTORY
cp [OPTION]... -t DIRECTORY SOURCE...
```

Esimerkki 15. Cp-komennon syntaksi (Computer Hope 2014.)

Yksinkertaisimmillaan komento voi olla muotoa: ”cp file.txt file\_.txt”. Tällöin kopioidaan tiedosto file.txt nimelle file\_.txt samaan hakemistoon. Toiseen kansioon kopioitaessa käytettäisiin komentoa: ”cp file.txt /tiedostopolku”. Tiedostopolkuun voidaan myös merkata hakemisto jota ei ole olemassa ja silloin se luodaan samalla. Kopioitavaa tiedostoa määrittäessä voidaan myös käyttää tiedostopolkua, jolloin ei tarvitse navigoida hakemistoon, jossa kyseinen tiedosto sijaitsee. Tiedostojen kopioimisessa voidaan käyttää useita lisäasetuksia, joita on listattu taulukkoon 5. (Computer Hope 2014.)

<b>i</b>	Tulostetaanvarmistuskehote, jos samanniminen tiedosto on jo olemassa.
<b>I</b>	Tekee kopioinnin sijasta kovia linkkejä.
<b>r</b>	Rekursiivinen kopiointi, kopioi myös hakemistot.
<b>t</b>	Määrittelee kohdehakemiston, kopioi kaikki muut määritellyt tiedostot tähän hakemistoon.
<b>u</b>	Kopioi vain, kun kohdetiedosto on vanhempi(muokkaus ajankohta) tai jos sitä ei ole.

Taulukko 5. Cp-komennon asetukset (Computer Hope 2014.)

Tiedostoja voidaan myös kopioida palvelimelta palvelimelle. Esimerkiksi kopioidaan tiedosto tai kansio testipalvelimelta tuotantopalvelimelle. Tällöin tulee olla kirjautuneen toiselle palvelimelle, omistaa tunnukset toisellekin palvelime ja tietää kansiorakenne sinne minne tiedosto halutaan kopioida.

Palvelimelta palvelimelle kopioinnissa käytetään **secure copy**:a eli **scp**-komentoa. Tämän komennon syntaksi on kuvattu yksinkertaistettuna esimerkissä 16. (Computer Hope 2014.)

```
scp [-o ssh_option] [-P port] [-S program] [[user@]host1:]file1 ... [[user@]host2:]file2
```

Esimerkki 16. Scp-komennon syntaksi (Computer Hope 2014.)

Eli jos haluamme kopioida tiedoston paikalliselta koneelta etäkohteeseen, käyttäisimme esimerkin 17. komentoa. (Computer Hope 2014.)

```
scp /kansio/kansio2/tiedosto.txt käyttäjä@kohteenOsoite:/kansio/kansio2
```

Esimerkki 17. Komento paikalliselta koneelta etäkoneelle kopioimiseen (Computer Hope 2014.)

Kopion voi myös ottaa paikalliselle koneelle etäkohteesta. Tällöin vain vaihdetaan kohteiden järjestys komennossa. Komentoon voi lisätä myös useita lisäasetuksia, joita on listattu taulukkoon 6. (Computer Hope 2014.)

<b>r</b>	Kopioi rekursiivisesti koko kansiorakenteen.
<b>q</b>	Kytkee edistymispalkin pois päältä.
<b>p</b>	Säilyttää alkuperäisen tiedoston kaikki ominaisuudet, kuten muokkausaika ja käyttöoikeudet.
<b>P</b>	Määrittää portin, jota käytetään yhdistäessä etäkohteeseen.
<b>v</b>	”Verbose mode” tulostaa scp:n ja ssh:n debugausviestit tehtävän etenemisestä.
<b>C</b>	Antaa ssh:lle luvan pakata dataa.

Taulukko 6. Scp-komennon asetukset (Computer Hope 2014.)



## 5 OHJELMISTOARKKITEHTUURI

Ohjelmistoarkkitehtuurin tarkoitus on määrittää järjestelmän osat, niiden keskinäiset suhteet ja niiden suhteet ympäristöön sekä periaatteet, jotka ohjaavat järjestelmän suunnittelua ja evoluutiota. (Koskimies & Mikkonen 2005.)

Arkkitehtuurin dokumentointi luo yhteisen ymmärryksen ohjelmistosta ja sen rakenteesta kehittäjien välille sekä helpottaa ohjelmiston ympärillä toimivien sidosryhmien välistä kommunikointia. Kun suunnitteluvaiheessa otetaan huomioon uudelleenkäytettävyys ja muunneltavuusvaatimukset saadaan työhön lisää tehokkuutta, koska valmista ohjelmakoodia voidaan käyttää myöhemmin uudelleen. Arkkitehtuuri dokumentaatio toimii järjestelmän yleissuunitelmana ja kertoo pääpiirtein, miten asioiden tulee toimia. (Seppänen 2010.)

Arkkitehtuuri dokumentaatioissa tulisi määrittää arkkitehtuurista riippuen esimerkiksi näkymät, käsittelijät ja mallit. Tärkeää on myös määrittää järjestelmän osien suhteet, eli miten mitkäkin näkymät, käsittelijät ja mallit liittyvät toisiinsa. Tämän lisäksi on oleellista tietää korkealla tasolla mikä tarkoitus ja millainen funktionaalisuus milläkin osalla on. (Seppänen 2010.)

Arkkitehtuurin määrittämät osat määritellään erilleen, jotta koodista saadaan helpommin hallittavaa ja luettavaa. Kun tiettyyn asiaan liittyvät toiminnallisuudet jaetaan omiin palasiinsa (separation of concerns) saadaan ohjelmakoodista modulaarisempaa ja täten uudelleenkäytettävyys paranee ja jatkokehitys on vaivattomampaa. (Koskimies & Mikkonen 2005.)

Ohjelmistoarkkitehtuureja voidaan implementoida käyttäen erilaisia työkaluja halutun arkkitehtuurin saavuttamiseksi. Otettakoon esimerkkinä järjestelmä, joka halutaan toteuttaa MVC-mallin mukaisesti käyttäen jQuery kirjastoa. Järjestelmässä on näkymä, käsittelijöitä ja malli. Näkymää muokataan dynaamisesti käsittelijöissä jQuery-kirjaston ominaisuuksia hyödyntäen. Näkymän ja käsittelijöiden välinen datan siirto tapahtuu JavaScript funktioiden avulla. Järjestelmän tietokanta on sidottu malliin, jolla on funktiot datan tallentamiseen ja hakemiseen tietokannasta. Malli tulee olla ohjelmoituna WWW-palvelimella esimerkiksi PHP-kielellä. Käsittelijästä voidaan tehdä malliin

funktiokutsuja AJAX:n avulla. Näin kontrolleri pystyy hakemaan ja tallentaamaan dataa tietokantaan mallin avustuksella.

## **5.1 Kykylaakson arkkitehtuuri**

Kykylaakson keskeisiä osia ovat TAMK:n kehittämät Portaali, Ainopankki-verkkopankkipalvelu, Verokarhu, Verkkokaupat ja Diileri-virtuaalitalouslehti. Lisäksi yksi keskeinen osa on Pupesoft, Devlab Oy:n kehittämä avoimenlähdekoodin toiminnanohjausjärjestelmä. Kykylaakson Pupesoft on modifioitu toimimaan yhdessä muiden kykylaakson osien kanssa ja kaikki osat käyttävät yhteistä tietokantaa. Lisäksi pupesoftiin on lisätty yrityspeli ominaisuuksia, joka mms. tekee viikoittain tilauksia oppiyrityksille. Kykylaakson osien keskinäiset suhteet ovat havainnollistettu visuaalisesti järjestelmä kaaviossa, joka on tämän dokumentin Liitteessä 1.

### **5.1.1 Kykylaakso tekninen dokumentaatio**

Johtuen kykylaakson ja varsinkin pupesoftin laajuudesta, ei näiden resussien puitteissa ole järkevää tai mahdollista tehdä kaiken kattavaa teknistä dokumentaatiota järjestelmästä, joka koostuu päälle kahdeksasta tuhannesta tiedostosta. Sen sijaan pyrin käymään pääpiirtein läpi kykylaakson toiminnan ja kertomaan sen tärkeimmistä komponenteista.

Kykylaakson eri osien toteutustavat eroavat hieman toisistaan sillä niitä on tehty eri henkilöiden toimesta vuosikymmenen saatossa. Suurimmat erot ovat kuitenkin TAMK:n osien ja pupesoftin välillä. Osien eroavista toteutustavoista huolimatta on koko järjestelmän toteutus vähintäänkin sekavaa ja vanhanaikaista. Korkealla tasolla järjestelmän arkkitehtuuria kuvaa parhaiten termi spagetti. Järjestelmän luokkarakennetta ei voi kuvata sillä sitä ei ole.

### 5.1.2 Pupesoft

Kuten jo aiemmin mainittiin niin myös pupesoftin arkkitehtuuria kuvaa parhaiten termi spagetti. Järjestelmässä ei ole minkäänlaista luokkarakennetta, eikä toiminnallisuutta ole jaoteltu omiin palasiinsa. Vain eri näkymät ovat erotettu toisistaan, mutta nekin vain eri tiedostoihin.

Istunnon tietojen käsittelyssä käytetään erillistä parametric.inc tiedostoa, johon päästään käsiksi ”require("inc/parametric.inc");” komennolla, niissä tiedostoissa joissa tarvitaan pääsy istunto tietoihin. Nykyaikana istuntoa hallittaisiin PHP:n omilla istunto ominaisuuksilla, joita ei varmaan vielä ollut silloin kuin Pupesoftia alettiin työstämään, eikä niitä ole myöhemmin päivitetty. Myös joitain usein käytettyjä PHP funktioita on jaettu .inc tiedostoihin.

Pupesoftin näkymät ovat staattisia html näkymiä, jotka generoidaan dynaamisesti PHP koodissa. Jokaiselle näkymälle on oma .PHP tiedostonsa, joka tulostaa pala palalta näkymää ja tekee tarvittaessa tietokantakyselyjä tai kutsuu.inc tiedostoja. Näkymät ovat linkitetty toisiinsa hyperlinkeillä, joilla viitataan suoraan .PHP tiedostoon tai sen funktioon, jolle välitetään URL:ssa tarvittaessa parametrejä. Linkin tiedostopolut tulevat usein tietokannasta ja tämä sekavoittaa koodin luettavuutta, sillä et voi koskaan suoraan tietää mitä tapahtuu.

Pupesoftin index.php-tiedosto alustaa istunnon ja jakaa näkymän kolmeen osaan HTML:n frame-tunnisteilla. Yhdessä osassa on aina pohjana indexvas.php. Joka tulostaa alasvetovalikkoon listan toiminnoista joihin käyttäjällä on oikeus. Lisäksi osaan tulostetaan dynaamisesti alamenulinkit, riippuen siitä mikä toiminto alasvetovalikossa on valittuna. Toinen osa on alamenu, joka näkyy vain harvoissa tilanteissa. Kolmas ruutu on pääruutu, jonka sisällön tulostaa oletuksena tervetuloa.php-tiedosto. Ruudun sisällöksi on määritetty URL joka on \$go-muuttujan arvo. Nyt sivuvalikosta valittaessa linkki, tallennetaan linkin URL \$go-muuttujaan ja ruudun pohja muuttuu dynaamisesti. Näkymän jaottelu on havainnollistettu kuvassa 8.

The screenshot shows a web application interface for 'KOPIOI ASIAKKAAN ALENNUKSIA'. The interface is divided into three frames:

- Left Frame:** Contains the 'indexvas.php' logo, 'Oriental Imports' branding, and a navigation menu with links like 'Ylikäyttäjä', 'Asiakkaat', 'Exit', 'Asiakasylläpito', etc.
- Top Frame:** Contains the search bar 'Etsi taulukosta:' and a login field 'tervetuloa.php'.
- Main Frame:** Contains a table of customer data with columns: Nimi, Nimitark, Postitp, and Ytunnus. The table lists various companies like Delowear, ErgoFurni Product, HäiveDesign, etc.

Nimi	Nimitark	Postitp	Ytunnus
Delowear		Kykylaakso	98144992
ErgoFurni Product		Kykylaakso	98144896
HäiveDesign		Kykylaakso	98144917
Konttorikatti Opy		Oivalla	98144263
LuxMappi		Kykylaakso	98144909
Oivamaatti Opy		Oivalla	98144319
OivaPaino Opy		Oivalla	98144327
OK-Huolto Opy		Oivalla	98144300
Present Bay Opy		Oivalla	98144343
Print my life		Kykylaakso	98144933
RemontOiva Opy		Oivalla	98144351
SensClean		Kykylaakso	98144888
SuperVerme Opy		Oivalla	98144255
Twisted binder		Kykylaakso	98144968

At the bottom of the table, it says 'Näytetään 1 - 14 / 14 rivistä' and 'Sanakirja'.

Kuva 8. Pupesoftin näkymän frame-jako

### Kolmannen osapuolen kirjastot

Pupesoft käyttää muutamia kolmannen osapuolen kirjastoja. Tällaisia kirjastoja käytetään ettei itse tarvitse koodata toiminnallisuutta, jonka joku muu on joskus koodannut ja jakanut yleiseen käyttöön. Näin säästetään paljon aikaa ja vaivaa. Kolmannen osapuolen kirjastot sisältävät yleensä joukon dokumentoituja funktioita, jotka helpottavat / mahdollistavat tietyn tyyppisen toiminnallisuuden luomista.

### Img\_Barcode

Img\_Barcode on Marcelo Subtil Marcalin kirjoittama .PHP luokka, jonka avulla on mahdollista generoida annetusta merkkijonosta viivakoodi kuvaformaattissa. Kykylaaksossa tätä käytetään laskujen viivakoodien generoimisessa.

Esimerkissä 18. yksinkertainen esimerkki Img\_Barcoden käytöstä. Ensin määritetään merkkijono josta viivakoodi generoidaan. Määritetään merkkijonon tyyppi esim. int tai string. Lopuksi määritetään minkä formaatin kuva luodaan ja syötetään tiedot Image\_Barcode luokan draw funktiolle.

```

<?php

require_once("Image/Barcode.php");

$num = isset($_REQUEST['num']) ? $_REQUEST['num'] : '15101967';
$type = isset($_REQUEST['type']) ? $_REQUEST['type'] : 'int25';
$imgtype = isset($_REQUEST['imgtype']) ? $_REQUEST['imgtype'] : 'png';

Image_Barcode::draw($num, $type, $imgtype);

?>

```

ESIMERKKI 18. `Img_Barcode` kirjaston käyttö

## PDFlib

PDFlib on Potential Technologiesille kuuluva kirjasto, jolla pystyy generoimaan .pdf formaatin tiedostoja. Kykylaaksossa PDFlibiä käytetään mms. laskujen ja kuittien generoimiseen. Samalla nimellä kulkee internetissä useita vastaavanlaisia kirjastoja, joista jokaista käytetään hieman omalla tavallaan. Tämän takia kirjaston käyttöön kannattaa etsiä ennemmin apua pupesoftin lähdekoodista ja PDFlib:n lähdekoodista kuin internetistä.

### 5.1.3 TAMK:n osat

TAMK:n osilla tarkoitetaan järjestelmän niitä osia jotka on toteutettu TAMK:n toimesta. Näihin osiin kuuluvat kykylaakso nettisivu eli portaali, ainopankki, verokarhu, diileri, tukiyritykset ja verkkokaupat. TAMK:n osien arkkitehtuuri poikkeaa pupesoftin arkkitehtuurista, sillä ne on toteutettu hieman nykyaikaisemmilla tekniikoilla. Myös TAMK:n eri osien arkkitehtuurit poikkeavat hieman toisistaan sillä ne ovat erillisiä palasia, jotka on tehty eri henkilöiden toimesta.

### Kykylaakso portaali

Portaali on siitä edistyksellinen kykylaakson osa että sillä on luokkarakenne. Portaalin pohjalla on `index` niminen abstrakti luokka, jonka jokainen näkymään tulostava luokka perii. `Index` luokan rakentaja kutsuu funktioita, jotka näyttävät näkymän eri

osat(otsikko, navigaatio, sisältö ja alatunniste). `Index` luokalla on abstrakti funktio `displayContent`, joka kirjoittaa sisältö-osion yli uuden pohjan. Koska `displayContent` on abstrakti luokka, on se pakko toteuttaa jokaisessa luokassa joka `index`:stä periytyy. Tulostusjärjestys eli kutsuttavat funktiot ja niiden järjestys määräytyy `index:n` rakentajassa. Jokainen näkymään tulostava luokka kutsuu rakentajassa perityn luokan rakentajaa, jossa `displayContent` luonnollisesti korvautuu luokkakohtaisella toteutuksella. Myös muut funktiot peritään oletuksena `Index`-luokasta, mutta nekin voidaan tarvittaessa korvata luokkakohtaisesti. Linkit sisältävät tarvittavat url-parametrit joiden perusteella määritetään mitä tulostetaan, missä muodossa ja tarvitaanko jatkotoimenpiteitä – vaikkapa kutsuja muihin funktioihin. Url-parametrien muuttujat on kovakoodattu PHP-tiedostoon joka generoi näkymän.

## **Ainopankki ja Verokarhu**

Ainopankin ja Verokarhun toimintatapa eroaa taas aiemmin kuvatuista kykylaakson osista. Näiläkään osilla ei ole luokkarakennetta ja näkymä generoidaan dynaamisesti eri puolilla PHP-koodia.

Istuntoa hallitaan PHP:n omilla istunto funktioilla niinkuin portaalissakin. Sisäämentäessä `index.php` luo istunnon ja tulostaa näkymään oletuksena otsikon, navigaatio palkin, pohjan ja sisällön, joka vielä tässä kohtaa on oletuksena sisäänkirjautumisruutu. Näkymässä ainoa muuttuva osa on sisältö osa, jonka sisältöä kontrolloidaan `content.php`-tiedostossa.

### **5.1.4 Ajoitetut tehtävät**

Toimiakseen kykylaakson tarvitsee suorittaa muutamia ajoitettuja tehtäviä. Ajoitettuihin tehtäviin kuuluvat lainanlyhennykset ja automaattitilaukset. Lainanlyhennykset ovat Ainopankissa kerran kuukaudessa tapahtuva toiminto, joka vähentää lainaerän verran rahaa jokaisen velkaa omaavan yrityksen tililtä. Tiedosto joka tämän suorittaa on ”`tarkistaLyhennys.php`” ja se sijaitsee ainopankki-kansion juuressa. Automaattitilaukset ovat yrityspelin ominaisuus, joka generoi kerran viikossa oppiyrityksille satunnaisia

tilauksia. Tämän toiminnon suorittavat tiedostot ”teeAutomaattitilaus.php” ja ”automaattitilaus.php” ja ne sijaitsevat pupesoftin yrityspeli-kansiossa.

Palvelimella ajoitetut tehtävät ovat ajastettu linuxin cronetab toiminnolla, jolla voi suorittaa määrättyjä tiedostoja määrätyn ajan välein. Tämä ei ole ratkaisusta paras sillä mahdollisen palvelimen vaihdon yhteydessä tarvitsee muistaa asettaa ajastukset uudelleen tai yrityspeli ei toimi.

## 6 YHTEENVETO

Kykylaakso on ympäristö, jota on kehitetty kymmenen vuoden ajan eri henkilöiden ja organisaatioiden toimesta. Tämän seurauksena käytetyt tekniikat ovat vanhanaikaisia eikä järjestelmä noudata yhdenmukaista tai järkevää arkkitehtuuria. Kykylaakson parissa työskennellessä oppii arvostamaan dokumentaation tärkeyttä.

Vaikkakin Kykylaakso on vanha järjestelmä, niin ei se riitä tekosyyksi sille miten epäammattimaisesti jotkin asiat on toteutettu. Suurimpana ongelmana on järjestelmän tietokanta ja se miten sitä käsitellään. Tietokannassa on enemmän monistunutta ja viittaamatonta dataa kuin oikeata käytettävää dataa. Tämä johtuu siitä että tietokannan viite-eheys on lähes olematonta eikä taulujen välille ole tehty oikeita relaatioita. Jos Kykylaaksosta esimerkiksi poistetaan yritys, suurin osa sen datasta jää turhaan tietokantaan. Toinen esimerkki Kykylaakson tietokannan kehnosta suunnittelusta on taulu jossa on käyttäjien käyttöoikeudet. Vaikka käyttäjiä on alle 50 niin taulussa on päälle 400 000 riviä dataa.

Nykypäivänä kykylaakson kaltainen järjestelmä toteutettaisiin hyvin eri tavalla. Web-sovelluksille on kehitetty useita hyviä arkkitehtuureja kuten MVC, MVVM ja MVW. Hyvällä arkkitehtuurilla saadaan suurenkin ohjelmiston lähdekoodista luettavaa, modulaarista ja järjestelmällistä. Kun näkymät, käsittelijät ja mallit erotetaan selvästi toisistaan on jatkokehityksin helpompaa. Nykypäivänä myös työkalut ovat parantuneet. Internet on pullollaan ilmaisia ja vapaastikäytettäviä kehitysympäristöjä, kirjastoja ja työkaluja. Halusitpa sitten itse implementoida tietyt työkalut haluamaasi arkkitehtuuriin tai löytää valmiin rungon niin joku on sen jo tehnyt ja siitä on tietoa internetissä.



## LÄHTEET

The World Wide Web Consortium. Introduction to HTML 4.0. Luettu 10.12.2013.  
<http://www.w3.org/TR/1998/REC-html40-19980424/intro/intro.html#h-2.2>

The World Wide Web Consortium. A history of HTML. Luettu 10.12.2013.  
<http://www.w3.org/People/Raggett/book4/ch02.html>

The PHP Group. History of PHP. Luettu 17.5.2013  
<http://PHP.net/manual/en/history.PHP.PHP>

Hudson, P. 2006. PHP in a nutshell. California: O'Riley Media Inc.

Jyväskylän yliopisto. PHP-sessiot. Luettu 14.4.2014.  
<http://users.jyu.fi/~kolli/ITK215/PHP/sessiot.html>

The World Wide Web Consortium. PHP Global Variables. Luettu 15.4.2014.  
[http://www.w3schools.com/php/php\\_superglobals.asp](http://www.w3schools.com/php/php_superglobals.asp)

Jyväskylän yliopisto. PHP-evästeet. Luettu 15.4.2014.  
[http://users.jyu.fi/~kolli/ITK215\\_05/php/?sivu=evasteet](http://users.jyu.fi/~kolli/ITK215_05/php/?sivu=evasteet)

Sitepoint. Native JavaScript Equivalents of jQuery Methods. Luettu 10.6.2013.  
<http://www.sitepoint.com/jquery-vs-raw-javascript-1-dom-forms/>

Wikipedia – Vapaa tietosanakirja. Tietokanta. Luettu 10.1.2014.  
<http://fi.wikipedia.org/wiki/Tietokanta>

Tuts+ www-sivut. SQL for beginners: Part 3. Luettu 14.1.2014.  
<http://net.tutsplus.com/tutorials/databases/sql-for-beginners-part-3-database-relationships/>

Wikipedia – The free encyclopedia. MySQL. Luettu 12.1.2014.  
<http://en.wikipedia.org/wiki/MySQL>

Koulutuskeskus Salpaus www-sivut. SQL-kieli. Luettu 14.1.2014.  
<http://edu.phkk.fi/Opiskelu/sqlkieli/default.htm>

New York University oppimateriaali. Database systems. Luettu 22.1.2014.  
<http://cs.nyu.edu/courses/spring01/G22.2433-001/mod4.2.pdf>

Linux.fi – wiki. Palvelin. Luettu 1.2.2014.  
<http://www.linux.fi/wiki/Palvelin>

Wikipedia – The free eyclopedia. Debian. Luettu 1.2.2014.  
<http://en.wikipedia.org/wiki/Debian>

Linuxcommand.org www-sivut. Navigation. Luettu 1.2.2014.  
<http://linuxcommand.org/lts0020.php>

Wikipedia – Vapaa tietosanakirja. Grep. Luettu 5.2.2014.

<http://fi.wikipedia.org/wiki/Grep>

Linux.org. Linux permissions – chmod. Luettu 6.2.2014.

<http://www.linux.org/threads/file-permissions-chmod.4094/>

Computer Hope. Linux and Unix grep command. Luettu 5.2.2014.

<http://www.computerhope.com/unix/ugrep.htm>

Computer Hope. Linux and Unix cp command. Luettu 6.2.2014.

<http://www.computerhope.com/unix/ucp.htm>

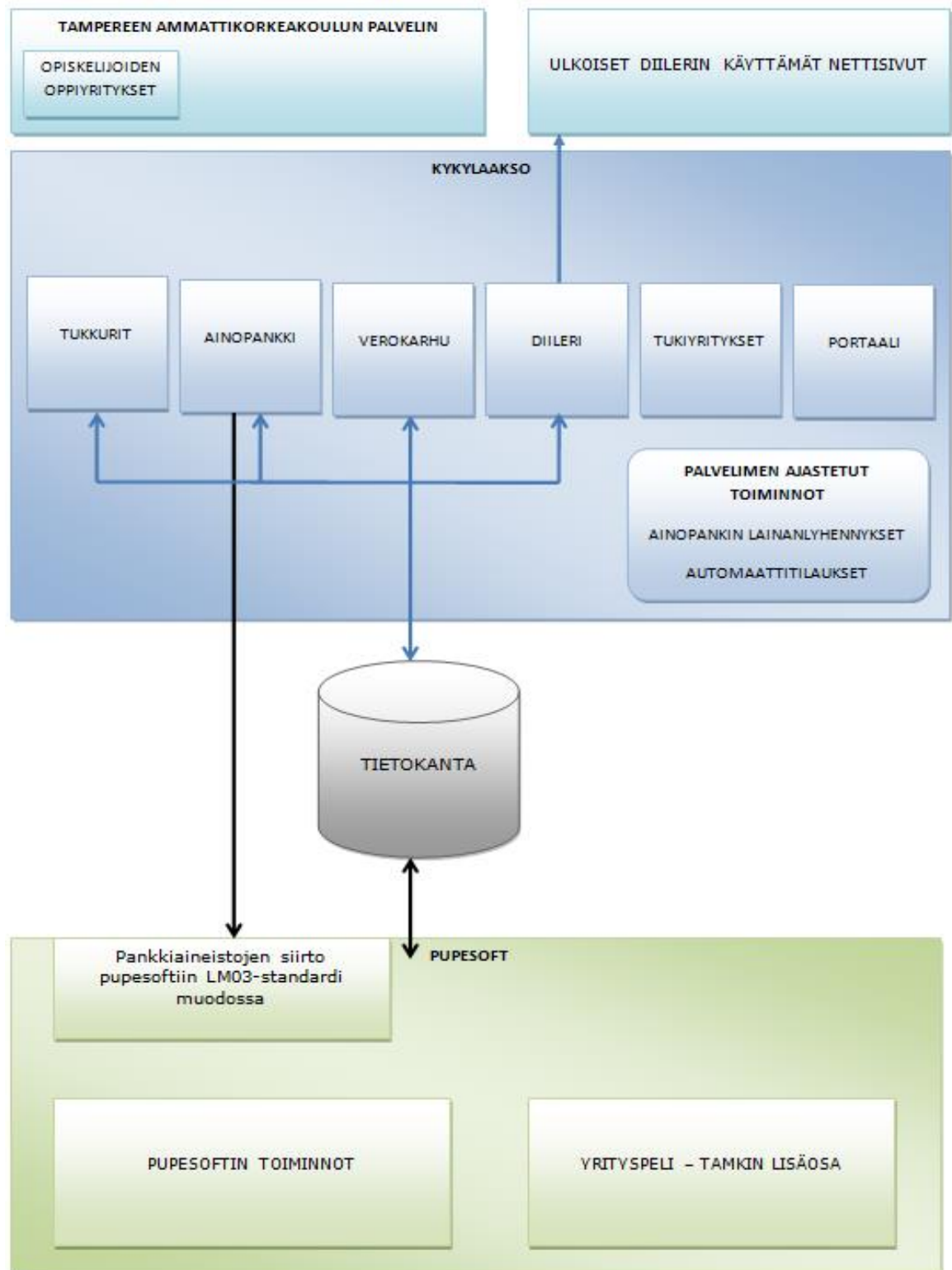
Koskimies, K., Mikkonen, T. 2005. Ohjelmistoarkkitehtuuri.

<http://cs.tut.fi/~kk/Ohjelmistoarkkitehtuuri.pdf>

Seppänen, J-T. 2010. Ohjelmistoarkkitehtuurit. Tietotekniikan koulutusohjelma. Saimaan ammattikorkeakoulu. Opinnäytetyö.

## LIITTEET

Liite 1. Kykylaakson järjestelmäkaavio (Kykylaakso dokumentit: Malli) 1(3)



## **KYKYLAAKSON PORTAALI**

WWW-portaali <http://www.kykylaakso.fi> on kykylaakson kaupungin web-sivusto joka sisältää tietoa kaupungista, sen yrityksistä ja linkityksen kykylaakson palveluihin.

## **DIILERI**

Kykylaakson uutissivusto

Sivuilla esitetään ulkoisilta nettisivuilta haettuja sää, valuuttakurssi ja uutistietoja

Käyttää tietokantaa

## **VEROKARHU**

Oppiyritykset voivat tehdä verokarhussa veroilmoituksia ja seurata tekemiensä ilmoitusten tilaa

Käyttää tietokantaa

## **TUKKURIT**

Kykylaaksossa on 3 tukkuria jotka ovat toiminnoiltaan identtisiä

Verkkokaupoissa voi tilata tavaraa virtuaalisesti Pupesoftin extranet-käyttäjätunnuksilla tavaraa, jotka toimitetaan laskun maksun jälkeen tilaavalle yritykselle

Verkkokauppa on osa pupesoftin toiminnallisuutta, joka on kopioitu kolmeen kertaan tukkureiksi kykylaakson ympäristön käyttöön

## **OPISKELIJOIDEN OPPIYRITYKSET**

Oppiyritykset ovat opiskelijoiden oppiyrityksilleen tekemiä nettisivuja. Sivut sijaitsevat Tampereen ammattikorkeakoulun palvelimella ja kykylaaksossa on vaan linkitys niihin. Sivuille ei ole mitään kykylaaksoon tai pupesoftiin vaikuttavaa toiminnallisuutta

## **TUKIYRITYKSET**

Tukiyritysten tehtävänä on tuoda todellisuudentuntua virtuaaliympäristöön. Web-sivujen ainoa tekninen toiminnallisuus on yhteydenottolomakkeet, joilla oppiyritykset voivat käydä sopimusneuvotteluita tukiyritysten kanssa.

## **YRITYSPELI / TILAUSAUTOMAATTI**

Tilausautomaatin tehtävä on automatisoida oppiyrityksille viikottain haluttu, syötteenä annettu määrä ostotilauksia esim. 3 ostotilausta/vko ja 5 tilausriviä/tilaus.

Tilausautomaatin käyttöliittymä on toteutettu pupesoftiin toiminnoksi kauppakeskus Myyrän sisälle, mutta teknisesti sen voisi myös irrottaa pupesoftista

Käyttää tietokantaa

## **AINOPANKKI**

Ainopankissa voi maksaa laskuja tai lainaa, sekä tarkastella yrityksensä rahatilannetta ja lainan määrää

Käyttää tietokantaa

## **TIETOKANTA**

Kykylaakso ja Pupesoft käyttää samaa yhteistä tietokantaa

## **PALVELIMEN AJASTETUT TOIMINNOT**

Palvelimella olevat ajastetut toiminnot hoitavat ainopankin lainanlyhennykset ja yrityspelin tilausten automatisoinnin

## **LM03 TIEDONSIIRTO**

Ainopankin pankkiaineistot siirretään pupesoftiin LM03-standardia noudattavien tiedostojen muodossa