

Liisa Peippo

## WEB-SISÄLLÖNHALLINTAJÄRJESTELMÄN SUUNNITTELU

Tietojenkäsittelyn koulutusohjelma

2015

# WEB-SISÄLLÖNHALLINTAJÄRJESTELMÄN SUUNNITTELU

Peippo, Liisa  
Satakunnan ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Helmikuu 2015  
Ohjaaja: Stenfors, Juha  
Sivumäärä: 90  
Liitteitä: 9

Asiasanat: avoin lähdekoodi, relaatiotietokannat, sisällönhallinta

---

Tässä opinnäytetyössä suunnitellaan web-sisällönhallintajärjestelmä nuorten harrastelijaylläpitäjien avuksi. Nuoret harrastelijaylläpitäjät on ryhmä, jolle on yhteistä kiinnostus jonkin muotoiseen luovaan taiteeseen sekä jonkinasteinen kokemus kotisivujen ylläpidosta. Näillä sivustoilla on usein samankaltaisena toistuvia rakenteita ja asioita, mutta toisaalta niillä voidaan myös kokeilla erilaisia asioita ja leikitellä luovilla ratkaisuilla.

Työn teoriaosiossa tutustutaan web-sisällönhallintaan yleisesti sekä Joomla!-sisällönhallintajärjestelmään ja phpBB-keskustelufoorumiin erityisemmin. Näiden lisäksi tutustutaan ohjelmistoprojektien teoriaan, tietokannansuunnittelun tärkeimpiin seikoihin ja käyttöliittymän, erityisesti web-käyttöliittymän, suunnitteluun. Lopuksi perehdytään hieman käyttöliittymän testaukseen.

Työn toiminnallisessa osassa tutustutaan erilaisiin nuorten harrastelijaylläpitäjien sivustojen tyypillisimpiin rakenteisiin sekä pohditaan järjestelmän tarvitsemää toiminnallisuutta. Suunnittelu aloitetaan itsenäisen järjestelmän tuottamiseksi, mutta lopulta päädytään luomaan järjestelmä phpBB-foorumin lisäosaksi. Jotta kohderyhmän tarpeet voidaan täyttää, on järjestelmän ensisijainen ominaisuus joustavuus.

# DESIGNING A WEB CONTENT MANAGEMENT SYSTEM

Peippo, Liisa

Satakunnan ammattikorkeakoulu, Satakunta University of Applied Sciences

Degree Programme in information technology

February 2015

Supervisor: Stenfors, Juha

Number of pages: 90

Appendices: 9

Keywords: open source, relational databases, content management

---

In this thesis, a web content management system is designed for young amateur web administrators. These administrators will typically have interest in some form of creative arts and have some amount of basic experience in administering a website. These websites usually have similar and/or repetitive structures and subjects, but the administrators can also test different things and play around with creative ideas there.

In the theoretical part of the thesis, web content management in general and the content management system Joomla! with the forum software phpBB in particular are examined. In addition, this section looks into the theory and concepts of software projects and the most important concepts of designing databases and user interfaces, especially web user interfaces. Lastly, there is a small explanation about testing the user interface design.

In the functional part of the thesis, the variety of the basic parts of young amateur administrators' websites are examined while the necessary system functionality is reflected upon. The design work is begun to develop a stand-alone system, but the system ends up being an add-on for a phpBB board. To fill the needs of the target group, the amateur administrators, the primary feature of the system is flexibility.

# SISÄLLYS

1	PROJEKTIN TAUSTAA .....	8
2	SUUNNITTELUN LÄHTÖKOHDAT .....	9
2.1	Mitä on Web-sisällönhallinta? .....	9
2.2	Kohderyhmä.....	11
2.3	Rajaus.....	11
2.4	Pohjana käytetyt järjestelmät .....	12
2.4.1	Joomla! .....	12
2.4.2	phpBB.....	13
3	SUUNNITTELU TEORIASSA .....	14
3.1	Projektien mallit.....	14
3.1.1	Vesiputousmalli.....	14
3.1.2	Prototyypilähestymistapa .....	15
3.1.3	Spiraalimalli .....	16
3.2	Järjestelmän määrittely .....	17
3.2.1	Vaatimusten ja tehtävien määrittely .....	17
3.2.2	Toiminnan määrittely .....	19
3.3	Relaatiotietokannan suunnittelu.....	20
3.3.1	Relaatiotietokanta lyhyesti .....	21
3.3.2	Kenttien määrittäminen .....	22
3.3.3	Taulujen määrittäminen.....	23
3.3.4	Pääavaimet .....	25
3.3.5	Viiteavaimet ja yhteydet.....	27
3.3.6	Denormalisointi .....	28
3.3.7	UML tietokannan kuvauksessa .....	30
3.4	Web-käyttöliittymän suunnittelu .....	32
3.4.1	Käyttöliittymäsuunnittelun yleiset piirteet .....	33
3.4.2	Sivuston latautumisen nopeus .....	35
3.4.3	Käytön ohjeistaminen.....	35
3.4.4	Intranet- ja ekstranet-tyyppiset järjestelmät .....	36
3.5	Käytettävyydestä .....	36
4	SUUNNITTELUN KULKU .....	38
4.1	Tehtävien määrittely .....	40
4.2	Ensimmäinen järjestelmä.....	47
4.2.1	Tietokannan suunnittelu .....	47
4.2.2	Käyttöliittymän suunnittelu.....	53
4.2.3	Käytettävyydestä .....	58

4.3 Toinen järjestelmä.....	59
4.3.1 Tyylien, käyttäjien ja oikeuksien hallinta .....	60
4.3.2 Sivuston asetusten hallinta .....	60
4.3.3 Lokitietojen keruu ja esittäminen.....	63
4.3.4 Tietokokonaisuuksien hallinta .....	64
4.3.5 Sisältöjen hallinta .....	70
4.3.6 Omien tietokenttien hallinta.....	75
4.3.7 Valikkojen hallinta .....	81
4.3.8 Etusivun hallinta.....	83
5 PROJEKTIN ONNISTUMISEN ARVIOINTI.....	85
LÄHTEET.....	89
LIITTEET	

## KÄYTETYT TERMIT

Banneri	Useimmiten kuvallisessa linkissä käytetty linkityskuva.
CMS	Content Management System, sisällönhallintajärjestelmä. Tätä ei kuitenkaan pidä sekoittaa lähteissä käytettyyn nimeen CMS, joka on lyhenne sanoista ”Centers for Medicare & Medicaid Services” ja kuuluu USA:n terveydenhuoltoon.
Ekstranet	Organisaation sisäiseen käyttöön tarkoitettu järjestelmä, johon voi olla yhteydessä myös Internetin kautta, ei siis pelkästään organisaation sisäisessä verkossa.
Elinkaarimalli	Malli, jonka perusteella tuotteen, tässä tapauksessa tietojärjestelmän, elinkaaren eteneminen kuvataan. Elinkaari alkaa jonkinlaisesta analyysistä ja päättyy tuotteen poistumiseen käytöstä, mutta se, mitä tässä välissä tapahtuu ja missä järjestyksessä, vaihtelee eri malleissa.
GPL-lisenssi	Lisenssi, joka mahdollistaa koodin vapaan muokkaamisen ja muokatun koodin julkaisun, kunhan koodin kerrotaan olevan muokatun alkuperäisestä ja kunhan näin muokattu tuote julkaistaan myös GPL-lisenssillä.
HTML-kieli	Web-selainten ymmärtämä kieli, jolla määritetään verkkosivujen rakennetta, ja jonka antamien määritysten pohjalta selain sitten näyttää sivun.
Intranet	Organisaation sisäisessä verkossa oleva, nykyään useimmiten web-sivuston kaltainen järjestelmä.

JavaScript	Web-sivuilla käytettävä ohjelmointikieli, jolla voidaan suorittaa toimintoja käyttäjän selaimessa, esimerkiksi asettaa osia sivusta näkyviksi ja piilottaa toisia ilman, että koko sivua täytyy ladata uudelleen.
Open Source	Ohjelmisto, jonka koodi on vapaasti käytettävissä ja muokattavissa. Open Source -periaatteella julkaistut ohjelmat on usein julkaistu GPL-lisenssillä, mutta toisinaan myös jollain muulla, samankaltaisella lisenssillä.
Sidosryhmä	Taho, joka on jollakin tavalla sidoksissa siihen asiaan josta puhutaan, tässä yhteydessä siis suunniteltavan järjestelmän erilaiset käyttäjät. Esimerkiksi ”henkilökunta”, ”asiakkaat” ja ”tavarantoimittajat” voisivat olla varastojärjestelmän sidosryhmiä, joilla on erilaiset tarpeet järjestelmän suhteen.
SQL	Kyselykieli, jolla kommunikoidaan tietokannan kanssa.
Tietokantaproseduuri	Tietokannan osa, ohjelma jolla voidaan suorittaa tietokannassa jokin toiminto.
URL-osoite	Osoite, jolla tietokone paikallistaa Internetissä olevat resurssit. Esimerkiksi se, mitä selaimen osoiteriville kirjoitetaan.
Validointi	Tiedon muodollisen oikeellisuuden tarkistaminen.
Web-hotelli	Palvelu, joka tarjoaa sivutilaa kotisivuille, useimmiten maksua vastaan. Web-hotelleja on kuitenkin tarjolla joka lähtöön, eivätkä kaikki ole maksullisia. Myös niiden tarjoamien palvelujen laajuus vaihtelee suuresti, vaikka se ei aina olekaan sidoksissa hintaan.

## 1 PROJEKTIN TAUSTAA

Tämän opinnäytetyön tavoitteena on suunnitella web-sisällönhallintajärjestelmä, jolla toteutettavan sivuston rakenne voidaan määrittää juuri sellaiseksi kuin ylläpitäjä tahtoo niin, että rakenne ei ole sidottu mihinkään valmiiksi määriteltyyn malliin. Vaikka valmiit mallit saattavatkin päteä useimpiin tilanteisiin, kaikkien sivustojen rakenteet eivät kuitenkaan sovi valmiisiin muotteihin, ja koska järjestelmällä on tarkoitus voida ylläpitää mahdollisimman monenlaisia sivustoja, rakenteen joustavuus on tärkeää.

Ajatus tähän projektiin lähti, kun etsin omaan käyttööni soveltuvaa sisällönhallintajärjestelmää. Löytämäni järjestelmät eivät vastanneet tarpeitani, sillä ne olivat aivan liian jäykkiä haluamieni sivustorakenteiden luomiseen, joten ainoaksi vaihtoehdoksi jäi suunnitella oma järjestelmä (tai vaihtoehtoisesti olla käyttämättä minkäänlaista järjestelmää ja jatkaa samalla kaavalla kuin siihenkin asti, kirjoittamalla kaiken käsin, mutta se käy pidemmän päälle puuduttavaksi ja etsin sisällönhallintajärjestelmää juuri siksi, koska olin kyllästynyt siihen valtavaan määrään käsityötä jota tämä tapa vaati). Tultuani tähän johtopäätökseen oli luonteva ajatus valita tämä myös opinnäytetyön aiheeksi.

Projekti lähtee liikkeelle tarpeiden selvittämisestä, mistä työskentely etenee muiden vastaavien järjestelmien tutkimiseen. Tästä lopputuloksena syntyy tarpeellisten toimintojen määrittely, jonka jälkeen suunnitellaan toiminnan mahdollistava tietokanta ja sitä käyttävä web-liittymä.

Tärkein toiminto on valmiiden rakenteiden hallinnointi ja niiden käyttäminen. Koska sivustoilla esiintyy usein toistuvia kaavaketyyppisiä rakenteita (esimerkiksi hahmosiivut), joiden kirjoittaminen uudelleen jokaiselle sivulle on sangen puuduttavaa puuhaa, järjestelmän tulee pystyä tällaisten rakenteiden määrittelyyn ja toteuttamiseen niin, että saman rakenteen käyttö usealla sivulla on vaivatonta. Rakenteet eivät kuitenkaan ole jokaisella sivustolla samanlaisia, ja samankin sivuston eri osissa ne voivat



poiketa toisistaan (esimerkiksi kahden eri tarinan hahmoista ei välttämättä esitellä samanlaisia tietoja), joten määrittelyjä on oltava mahdollista myös muokata.

Rakenteiden määrittelyn lisäksi sivuston hierarkian hallinnoinnin joustavuus on olennainen osa järjestelmää, jotta ylläpitäjää ei kahlita mihinkään jäykkään sivustorakenteeseen, vaan halutunlaisten, monipolvisten alikategoriajärjestelmien luominen on mahdollista. Linkkien muodostaminen helposti muihin sivuston osiin myös sivun sisältöosassa, ei pelkästään valikoissa, on sivuston rakenteen hallinnoinnin kannalta erittäin tärkeää, joten sen on oltava tärkeää myös järjestelmässä. Kaikkia linkkejä sivuston eri osiin ei välttämättä haluta nimenomaan valikkoon.

Oleennaista on tietysti myös, että näin luotu kokonaisuus voidaan rakentaa valmiiksi sivustoksi ja sitten näyttää selaajalle, tosin tämä kuuluu jo enemmän järjestelmän varsinaisen toteuttamisen kuin sen suunnittelun vaiheisiin, eikä sen ajattelu näin ollen ole välttämättä olennainen osa itse suunnittelua. Tietyiltä osiltaan sitä on kuitenkin syytä tutkia myös suunnitteluvaiheissa, erityisesti käyttöliittymän kohdalla, joten työssä sivutaan myös sivuston esittämiseen johtavia seikkoja.

## 2 SUUNNITTELUN LÄHTÖKOHDAT

### 2.1 Mitä on Web-sisällönhallinta?

Laajimmassa merkityksessään sisällönhallinta tarkoittaa kaikkea sisällön ylläpitoa sen tuottamisesta julkaisemiseen, arkistointiin ja poistamiseen. Sisällönhallinnan tarpeessa on kysymys nimenomaan sisällöstä, ei sen julkaisutekniikasta, vaikka hallinta onkin mahdotonta ilman jonkinlaista tekniikkaa. Sisällönhallinta sinällään kattaa kaikki esimerkiksi yrityksessä tarvittavat dokumentit, liiketoiminnan viralliset asiakirjat, muistiot ja muut yrityksessä käytössä olevat asiakirjat. Sisällönhallinta voidaan jakaa kolmeen osaan: virallisiin dokumentteihin, toimistodokumentteihin ja verkkosisältöön. (Samela 2002, 7-9, 13-14) Varhaiset sisällönhallintajärjestelmät tehtiin hallinnoimaan dokumentteja ja tiedostoja paikallisesti tietokoneella, nykyisiä käytetään pääasiassa sisällönhall-

lintaan verkossa. (Kohan 2010; Techterms.com, CMS) Voidaan siis todeta, että nykyään kaikki, tai ainakin miltei kaikki, sisällönhallinta on verkkosisällönhallintaa.

Verkko- eli web-sisällönhallinta on intranet-, ekstranet- ja Internet-sivuilla esitettävän sisällön ja siinä tarvittavien dokumenttien hallintaa. Web-sisällönhallinnassa ei hallinnoida niinkään erillisiä dokumentteja kuin toisiinsa linkitettyjä sivuja ja niiden muodostamia rakenteita. (Samela 2002, 14) Tähän asian selkeys sitten päättyykin, sikäli kuin se on ollut selkeää tähänkään asti. Bernard Kohanin (2010) mukaan edes sana ”sisällönhallintajärjestelmä” ei ole yksiselitteinen: sillä voi olla eri merkityksiä sen mukaan, missä yhteydessä sitä käytetään ja kuka termiä käyttää. Sekavuutta lisää vielä se, että AIM-niminen organisaatio on yrittänyt luoda sisällönhallintajärjestelmälle määritelmän, joka soveltuu kyseisen yrityksen tarpeisiin, mutta poikkeaa yleisesti käytössä olevista määritelmistä.

Kun joku puhuu www-ympäristössä sisällönhallintajärjestelmästä, kyseessä on todennäköisesti toinen kahdesta eri vaihtoehdosta: www-pohjaisesta järjestelmästä, jolla ryhmä erilaisilla käyttöoikeuksilla varustettuja käyttäjiä voi hallinnoida www-sivuston sisältöä ilman tietämystä www-sivujen luontiin tarvittavasta HTML-kielestä, tai ohjelmistosta, joka todennäköisesti on www-pohjainen mutta ei välttämättä ole sitä, ja jolla ryhmä erilaisilla käyttöoikeuksilla varustettuja käyttäjiä voi hallinnoida kaikkea tai osaa www-tai intranet-sivun sisällöstä tai tiedoista. (Kohan 2010)

Sisällönhallintajärjestelmän tarkoituksena on tarjota intuitiivisesti käytettävä käyttöliittymä, jonka avulla voi rakentaa ja muokata www-sivun sisältöä. Käyttäjän antamien tietojen ja sisällönhallintajärjestelmään ohjelmoidun koodin perusteella järjestelmä sitten rakentaa sivuston. Niitä on saatavilla sekä erillisinä ohjelmina että www-käyttöliittymillä varustettuina, joista jälkimmäiset ovat huomattavasti yleisemmässä käytössä siksi, että www-sivun hallinnointi koetaan helpommaksi jos jo ollaan www-sivulla. (Techterms.com, CMS) Myös tässä projektissa suunniteltava järjestelmä edustaa jälkimmäistä tyyppiä, koska se on käyttäjälle helpoin asennettava.

## 2.2 Kohderyhmä

Järjestelmän tarkoituksena on soveltua monenlaisten sivustojen ylläpitoon, mutta sen ensisijaisena kohderyhmänä ovat nuoret, ja mikä ettei vanhemmatkin, harrastelijaylläpitäjät. Tämän ryhmän tarpeet poikkeavat esimerkiksi yritysten sisällönhallintajärjestelmilleen määrittämistä tarpeista niin paljon, että sopivien termien ja rakenteiden määrittäminen vaatii keskittymistä juuri tähän ryhmään, mikä ei tietenkään tarkoita etteikö järjestelmä voisi sopia käytettäväksi myös muiden kuin tähän ryhmään kuuluvien henkilöiden sivustoilla.

Nuorten harrastelijaylläpitäjien sivustot sisältävät usein tarinoita, kuvia ja niissä esiintyvien hahmojen esittelyjä sekä luetteloita muille sivustoille johtavista linkeistä. Kohderyhmän tarpeiden tyydyttämiseksi on siis syytä mahdollistaa järjestelmällä ainakin näiden sisältöjen käsitteleminen. Lisäksi sivustoille on syytä pystyä lisäämään muutakin sisältöä, sillä harrastelijaylläpitäjien sivustojen sisältö ei kuitenkaan rajoitu pelkästään näihin mainittuihin asioihin, ja ylläpitäjät mielellään laajentavat sivustojaan ajan kuluessa ja taitojensa karttuessa.

## 2.3 Rajaus

Suunniteltava järjestelmä sisältää ulkoasujen, rakenteiden, sisällön ja sivuston asetusten hallinnan. Lisäksi järjestelmän on mahdollistettava sisällön esittäminen, sillä sisällön hallinnasta on hyötyä vain, jos se myös esitellään.

Ulkoasujen hallinta kattaa kansioon ladattujen ulkoasujen käyttöönoton ja käytöstä poistamisen. Rakenteiden hallinnassa ylläpitäjälle tarjotaan mahdollisuus käyttää valmiita rakenteen pohjia sisällön hallintaan ja esittämiseen tai luoda omia rakenteita. Kutsun näitä rakenteita vastedes nimellä rakennepohjat.

Sisällön hallinnassa rakennepohjien avulla lisätään sivustolle sisältöä, joka sitten koostetaan sivuiksi ylläpitäjän määrittämällä tavalla. Sisältöä esitettäessä tallennettu sisältö esitetään ulkoasun määrittämällä asettelulla rakennepohjan tai -pohjien määrittämällä tavalla.

Asetusten hallinnassa hallinnoidaan sivuston yleisiä asetuksia, kuten sivuston nimeä ja käytössä olevaa ulkoasua. Tähän kuuluvat myös esimerkiksi ylläpitäjän yhteystiedot ja hallintapaneelin kirjautumistiedot. Nämä asetukset koskevat koko sivustoa.

Rajaus ei sisällä käyttäjien ja näiden oikeuksien hallintaa, sillä tämä järjestelmä mahdollistaa vain yhden ylläpitäjän eikä anna rekisteröitymismahdollisuutta sivustolle. Rajaus ei sisällä myöskään useiden sivustojen hallintaa, joten järjestelmällä voi hallinnoida vain yhtä sivustoa kerrallaan.

Suunniteltava järjestelmä on työnimeltään ACMS. Käytän paikoitellen tätä nimeä tässä raportissa, toisin paikoin kutsun sitä yksinkertaisesti ”tässä projektissa suunniteltavaksi sisällönhallintajärjestelmäksi” tai jollain vastaavalla nimityksellä.

## 2.4 Pohjana käytetyt järjestelmät

Perehtyäkseni web-sisällönhallintaan käytännössä tutustuin mainittuun tarkoitukseen suunniteltuihin ilmaisiin järjestelmiin. Kohdistin huomioni ilmaisiin järjestelmiin, koska budjettini eikä suoranaisesti kiinnostuksenikaan riittänyt kaupallisten järjestelmien hankkimiseen ja testaamiseen, varsinkin kun ilmaiset vaikuttivat tässä yhteydessä täysin riittäviltä. Lisäksi General Public- eli GPL-lisenssillä julkaistujen ilmaisten järjestelmien käytössä on se etu, että niiden koodi on vapaasti käytettävissä, kunhan lähde mainitaan ja kunhan valmis tuote julkaistaan myös GPL-lisenssillä, mikäli tuote siis aiotaan myös tarjota muiden saataville. Suunnittelussa olen käyttänyt pohjana pääasiassa kahta tällaista järjestelmää, joista toinen on Joomla! ja toinen phpBB. (Kopio näiden kahden käyttämästä GPL 2 -lisenssistä löytyy liitteestä 1.)

### 2.4.1 Joomla!

Joomla! on web-sisällönhallintajärjestelmä, joka on tarkoitettu web-sivustojen rakentamiseen ja hallintaan. Järjestelmä on toteutettu Open Source -periaatteella, joten se on vapaasti kaikkien saatavissa. Järjestelmän käyttö on hyvin suosittua, ja sen oman esittelytekstin mukaan sitä on myös helppo käyttää. (Joomla.org, What is Joomla?)

Oman kokemukseni mukaan Joomla!-järjestelmällä sivuston sisällön ja ennen kaikkea rakenteen hallinnointi on kaikkea muuta kuin helppoa, vaikka esittelytekstissä toisin väitetäänkin. Osasyllisenä saattaa tosin olla se, että en ole järjestelmää juurikaan käyttänyt, sillä tutustuin siihen pääasiassa tämän projektin vuoksi. Kenties käyttäminen olisi helpompaa, jos olisin siinä harjaantuneempi, mutta ainakin käytön aloittaminen ja järjestelmän toimintalogiikan ymmärtäminen on haasteellista. Käyttöliittymä ei siis ole erityisen intuitiivisesti hahmotettavissa.

Joomla!n valitsin, koska se oli tosiasiasa ainoa sisällönhallintajärjestelmä joka palkähti päähäni ilman sen kummempaa tutkimustyötä. Siinä on myös se etu moniin kaupallisiin järjestelmiin verrattuna, että se on julkaistu GPL-lisenssillä, joten vaikutteiden ottaminen siitä ei aiheuta juridisia ongelmia. Tämän järjestelmän oli alun alkaenkin tarkoitus vain toimia referenssinä sille, millaisia toimintoja sisällönhallintajärjestelmän tulisi sisältää, ja sen asian Joomla! nähdäkseni hoitaa siinä missä mikä muu sen virkaveljistä tahansa.

#### 2.4.2 phpBB

phpBB on keskustelufoorumiohjelmisto, jonka käyttö on levinnyt laajalle ympäri maailman. Ohjelmisto on toteutettu Open Source -periaatteella, ja se on julkaistu GNU General Public License -lisenssillä, mikä tarkoittaa, että ohjelmakoodia jaetaan vapaasti ja sen muokkaaminen on sallittua. (phpBB Group, About phpBB)

Vaikka kyseessä onkin keskustelufoorumi, phpBB sisältää monenlaisia ominaisuuksia kuvien ja sisällön hallintaan, varsinkin jos sitä hieman sovelletaan. Ylläpitoliittymän kautta voidaan hallinnoida monia sellaisia asioita, joita web-sivustojenkin sisällönhallinnassa tarvitaan. Lisäksi olen itse ylläpitänyt phpBB-foorumia useita vuosia, joten ympäristö on entuudestaan tuttu, mikä helpottaa järjestelmän ominaisuuksien ja toiminnallisuuksien tutkimista ja ymmärtämistä ilman pitkäkestoista opettelua. Sekään ei tosin ole käytön aloittamisen suhteen erityisen intuitiivinen, vaikka kokeneena käyttäjänä tätä tosiasiaa onkin välillä varsin vaikea muistaa.

phpBB:n valitsin sillä, vaikka se ei olekaan varsinaisesti sisällönhallintajärjestelmä, omien kokemusteni vuoksi sen toimintaperiaate on itselleni kaikkein tutuin ja helpoimmin lähestyttävä. Se on myös joustavuudeltaan kaikkein lähinnä sitä, mitä omalta järjestelmältäni hain. Kun työn edetessä kävi ilmeiseksi, että itsenäisen järjestelmän sijaan olisi kannattavampaa tehdä järjestelmästä lisäosa keskustelufoorumille, oli jo täysin selvää, että alustaksi valittavana keskustelufoorumina toimisi juuri phpBB.

### 3 SUUNNITTELU TEORIASSA

#### 3.1 Projektien mallit

Koska laajojen tietojärjestelmien kehittäminen on monimutkaista, on niiden hallinnoinnin apuvälineiksi kehitetty erilaisia malleja. Näitä malleja kutsutaan muun muassa elinkaarimalleiksi ja systeemikehitysmetodeiksi. (CMS 2005; Pohjonen 2002, 39; Ruohonen & Salmela 2005, 76) Ne ovat silti pelkkiä malleja, jotka eivät sinällään kuvaa minäkään yrityksen toimintaa, vaan ainoastaan sitä miten voidaan toimia. Jokainen organisaatio voi tulkita niitä hieman omalla tavallaan, jossa periaate on sama mutta yksityiskohdat vaihtelevat. (Kasurinen 2013, 29)

Malleja on olemassa muitakin kuin tässä esitettävät, mutta valitsin tarkemmin tarkasteltaviksi ne, jotka löysin useimmista lähteistä, koska arvioin ne vakiintuneimmiksi.

##### 3.1.1 Vesiputousmalli

Vanhin ja vakiintunein näistä malleista on vesiputousmalli. Se kehitettiin jo 1960-luvulla perinteisempien teollisuudenalojen prosessimallien pohjalta. Vesiputousmallissa ohjelmistoprojektin vaiheet seuraavat toisiaan siten, että yksi vaihe suoritetaan aina kerrallaan, ja kun se on suoritettu ja hyväksytty, siirrytään seuraavaan. Varhaisempiin vaiheisiin ei ole tarkoitus juuri palata, ja tällainen peruuttaminen onkin vesiputousmallissa hyvin vaikeaa. (Kasurinen 2013, 22; Pohjonen 2002, 40)

Vesiputousmalli toimii periaatetasolla loistavasti, mutta ohjelmistotuotannon arki on ikävä kyllä harvoin yhtä yksinkertainen. Varhaisemmissa vaiheissa tehtyjä virheitä paljastuu usein vasta myöhempien vaiheiden suorittamisessa, mikä aiheuttaa kalliita korjaustoimenpiteitä, koska edeltäviin vaiheisiin tehdyt korjaukset aiheuttavat tarpeen suorittaa koko vaiheet uudelleen. Lisäksi ongelmana on, että mallin alussa suoritettavat analyysit ja määrittelyt harvoin kattavat koko totuutta järjestelmästä, jota ollaan tekemässä. (CMS 2005; Kasurinen 2013, 22-23; Pohjonen 2002, 40)

Vesiputousmallin vahvuudet ovat sen kontrolloitavuudessa. Se tuottaa ennakoitavia ja mitattavia aikatauluja, ja sen etenemistä ja aikataulussa pysymistä on helppoa ja mahdollista sekä seurata että mitata. Vähemmän kokeneille projektiryhmille vesiputousmalli on hyvä lähtökohta. Se sopii käytettäväksi parhaiten suuriin, kalliisiin ja monimutkaisiin projekteihin, joissa tarpeet ja vaatimukset voidaan luotettavasti määrittellä heti projektin alussa. Lisäksi se on hyödyllinen silloin, kun projektin henkilöstö vaihtuu projektin aikana tai projektin johdolla on vähänlaisesti kokemusta projektien, erityisesti tietojärjestelmäprojektien, johtamisesta. (CMS 2005)

### 3.1.2 Prototyypilähestymistapa

Vesiputousmallissa syntyy varsin myöhään mitään konkreettista asiakkaalle esitettäväksi, ellei oteta huomioon suunnittelussa syntyneitä dokumentaatiota. Jos asiakas kuitenkin tahtoo päästä kokeilemaan tuotetta jo mahdollisimman varhaisessa vaiheessa, tai jos tarpeiden määrittely on alkuvaiheessa hankalaa, ei vesiputousmallista ole juuri apua. Tällaisia tilanteita varten on kehitetty prototyypilähestymistapa, jossa alussa tehdyn analyysin pohjalta rakennetaan prototyyppi. Tällaisessa prototyypissä on tyypillisesti järjestelmän karkea toiminnallisuus, mutta ei yksityiskohtia. Asiakas kokeilee prototyyppiä, ja palautteen pohjalta rakennetaan uusia prototyyppijä, kunnes asiakas on tyytyväinen. Varsinainen järjestelmä rakennetaan sitten tämän viimeisen prototyypin pohjalta. (Pohjonen 2002, 40-42)

Prototyypilähestymistapa ei sinällään ole kuitenkaan välttämättä itsenäinen malli, vaan pikemminkin tapa suorittaa osioita suuremmasta, jollakin perinteisemmällä metodilla

toteutetusta projektista. Se ei myöskään ole täysin ongelmaton: tarve tuottaa toimintakuntoisia prototyyppijä saattaa aiheuttaa moninkertaista työtä, eivätkä kaikki järjestelmän heikkoudet välttämättä tule ilmi tällaisessa mutkia oikovassa lähestymistavassa. On mahdollista, että suunnittelijat työstävät järjestelmää puutteellisten ja jopa virheellisten tietojen perusteella. (CMS 2005; Pohjonen 2002, 42)

Prototyypilähestymistapa soveltuu parhaiten tilanteisiin, joissa projektin tavoitteet eivät ole täysin selvät, projektiryhmä on kokenut ja rakenteeltaan vakaa, ja järjestelmän vaatimukset voivat muuttua paljon lyhyessä ajassa, erityisesti jos järjestelmän tulee kommunikoida runsaasti käyttäjiensä kanssa. Jos varsinainen järjestelmä rakennetaan käyttäen hyväksi prototyyppijä, sen sijaan että siinä käytettäisiin vain prototyyppien toimintaperiaatteita, projektin jäsenten kokeneisuus on erityisen tärkeää, jotta järjestelmästä saadaan yhtenäinen. (CMS 2005)

### 3.1.3 Spiraalimalli

Spiraalimallin pääajatuksena on, että projektia tuotetaan ja arvioidaan sykleittäin. Se koostuu neljästä vaiheesta (suunnittelu, riskianalyysi, tuotanto, asiakkaan suorittama arviointi) joita toistetaan kuin kierroksia spiraalissa, kunnes projekti on valmis, tai keskeytetty mikäli riskit todetaan jollakin kierroksella liian suuriksi. Jatkuva riskien arviointi on jotakin, joka on jäänyt huomattavasti vähemmälle huomiolle muissa malleissa, mutta on korostetusti esillä spiraalimallissa. (CMS 2005; Pohjonen 2002, 42-43)

Spiraalimalli on varsin uusi lähestymistapa, eikä se varsinaisesti määritä millä menetelmällä työvaiheet tulisi suorittaa: jokainen spiraalin kierros voidaan esimerkiksi suorittaa omana vesiputouksenaan. Tai, kun projektin riskiarvio muuttuu uuden kierroksen alkaessa, voidaan työskentelytapaa muuttaa kulloiseenkin tilanteeseen sopivaksi: yksi kierros voidaan suorittaa vesiputouksena, toinen prototyyppijä tekemällä. Oikean työskentelytavan valinta voi kuitenkin olla vaikeaa, ja koska spiraalissa ei oikeastaan ole tiukoja kiinnekohtia vaiheesta toiseen siirtymiselle, se on menetelmänä varsin monimutkainen. (CMS 2005; Pohjonen 2002, 43)



Spiraalimalli on omimmillaan tilanteissa, joissa riskienhallinta on ensisijaisen tärkeää syystä tai toisesta. Jos vielä projekti voi hyötyä siitä, ettei sen tarvitse joka vaiheessa sitoutua yhteen ja samaan lähestymistapaan, ovat spiraalimallin edut ilmeiset. Se vaatii kuitenkin toimiakseen erittäin taitavan ja kokeneen projektipäällikön sekä tarkat vaatimukset siitä, missä vaiheessa milloinkin ollaan ja tulisi olla. (CMS 2005)

### 3.2 Järjestelmän määrittely

Järjestelmän suunnitteluprosessi alkaa tarpeiden kartoittamisella, jolloin selvitetään mitä järjestelmältä odotetaan ja mihin sitä tarvitaan. Kartoituksessa saadun tuloksen perusteella määritellään järjestelmän tehtävät. Tässä vaiheessa määritellään järjestelmän yleinen tarkoitus, joka toimii perustana koko järjestelmän suunnittelulle. Tarkoitustaan tukeva järjestelmä on mahdollista rakentaa vain, jos sen tarkoitus tunnetaan. Lisäksi määritellään ne tehtävät, joita järjestelmällä tulee pystyä suorittamaan. (Hernandez 2000, 61-62; Samela 2002, 209-210)

#### 3.2.1 Vaatimusten ja tehtävien määrittely

Ensiksi on tehtävä vaatimusmäärittely, johon kootaan eri sidosryhmien vaatimukset siitä, mitä järjestelmän tulee tehdä ja missä olosuhteissa sen tulee toimia. Vaatimusten määrittäminen on vaikeaa, sillä niiden keräämiseen ei ole olemassa yhtä ainoaa oikeaa menetelmää, joten operaatiossa on mahdollisesti sovellettava useita erilaisia menetelmiä. Asiakkaat eivät välttämättä osaa antaa tarkkoja kuvauksia siitä, mitä järjestelmältä oikeastaan haluavat, vaatimukset voivat olla keskenään ristiriitaisia tai yksinkertaisesti täysin järjettömiä. Toisinaan myös mielekkäältä vaikuttavan vaatimuksen takana voi olla järjetön ajatus, mutta toisaalta järjettömän vaatimuksen takana voi olla mielekäs logiikka. Määrityksen tekijöiden tehtävänä on saada selkoa tästä kaaoksesta ja tuottaa siitä looginen kokonaisuus, joka täyttää riittävän suuren osan sidosryhmien mielekkäistä tarpeista järjestelmän suhteen. (Pohjonen 2002, 28-30)

Määritelmien laatimiseksi tulee opetella organisaation toimintaa sekä haastatella sen henkilöstöä ja johtajia. Käyttökelpoisia ovat myös erilaiset ideointipalaverit ja markkinatutkimukset. Vain järjestelmän tilaajat voivat tietää, mihin tarkoitukseen se on tilattu,

ja vain järjestelmän tulevat käyttäjät voivat tietää, mitä tietoja ja toimintoja he tarvitsevat työssään. Organisaation toiminnan tukemiseksi on saatava selville, miten organisaatio toimii ja mikä on sen tarkoitus. (Hernandez 2000, 83-84, 89; Pohjonen 2002, 29)

Järjestelmän yleistä tarkoitusta selvittäessä tulee haastatella joko organisaation johtajaa tai erikseen tähän tarkoitukseen soveltuvaksi nimettyä henkilökunnan jäsentä. Haastateltavan tulee tuntea organisaatio ja tietää, mihin se tarvitsee järjestelmää. Tehtävien määrittämiseksi on syytä haastatella järjestelmän tulevia käyttäjiä, jotta saadaan selville, mitä nämä järjestelmältä odottavat ja tarvitsevat. Haastatteluja tehdessä on parasta käyttää avoimia kysymyksiä, jotka koskevat yrityksen ja työntekijöiden tehtäviä yleisesti, ja kysyä tarkentavia kysymyksiä, jos vastauksista ei saa kaikkea tarvittavaa tietoa. (Hernandez 2000, 83-84, 89-91)

Web-sisällönhallintajärjestelmän toimintoja määrittäessä on selvittävä minkälaista sisältöä on tarkoitus hallita, kuinka monimutkaisia linkkirakenteita on odotettavissa, millainen olisi helppo käyttöliittymä ja voidaanko sisältö jakaa sellaisiin joukkoihin, joiden välillä ei ole paljon linkkejä. Lisäksi on selvittävä näytetäänkö kaikille käyttäjille sama sisältö, kuinka monenlaisissa muodoissa sisältö on tarkoitus esittää ja millä tavalla sisältö tallennetaan. On tarpeen tutkia myös kuinka paljon henkilöitä osallistuu sisällön tuottamiseen, mitä vaatimuksia sisällön laadulle ja visuaaliselle ilmeelle asetetaan ja kuinka paljon aineistoa on. (Samela 2002, 38-39, 60, 218)

Haastatteluista saadun aineiston pohjalta kirjoitetaan kuvaukset järjestelmän tarkoituksesta ja tehtävistä. Kuvausten tulee olla lyhyitä ja selkeitä, ja yhden kuvauksen on kuvattava vain yhtä asiaa. Kirjoitustyön valmistuttua kuvaukset tulee tarkastuttaa organisaation henkilöstöllä. Tarkastuksen tarkoituksena on selvittää, onko jokin tavoite jäänyt havaitsematta, ja muotoilla kuvaukset sellaisiksi, että kaikki voivat hyväksyä ne. Kun tarkastus on suoritettu, ja kuvaukset tyydyttävät kaikkia, tehtävien määrittely on valmis. (Hernandez 2000, 81, 87-89, 93)

### 3.2.2 Toiminnan määrittely

Kun vaatimukset on määritelty, voidaan siirtyä itse järjestelmän määrittämiseen. Tässä vaiheessa määritetään, miten järjestelmä toimii jotta se voi täyttää ne vaatimukset, joita järjestelmälle on asetettu. Nämä määrittelyt eivät kuitenkaan koske varsinaisesti itse käytännön toteutusta (esimerkiksi käytettävää ohjelmointikieltä), vaan pikemminkin sitä, millä tavalla järjestelmän tulisi käyttäytyä. Määrittelyt kuvaavat siis, mitä järjestelmä tekee, ja osittain myös sen, miten sitä käytetään. (Pohjonen 2002, 31-32)

Toimintaa määriteltäessä yleensä tuotetaan enemmän tai vähemmän valtaisa määrä erilaisia kaavioita, joilla mallinnetaan tilaajan ja järjestelmän toimintoja (tilaajan siinä määrin kuin se on järjestelmän kannalta olennaista). Voidaan esimerkiksi tehdä prosessikaavioita, joissa kuvataan ne toimenpiteet, joilla järjestelmän tulevat käyttäjät ovat aiemmin suorittaneet toiminnon, joka tulee nyt suunniteltavan järjestelmän suoritettavaksi. Voidaan myös tehdä tietovirtakaavioita, joista nähdään mitä tietoa järjestelmän tulee välittää mihinkin suuntaan. On myös tilamalleja, joilla kuvataan järjestelmän tai sen osan tilaa eri aikoina, ja sekvenssikaavioita, joilla kuvataan järjestelmän eri osien keskinäistä kommunikaatiota. (Pohjonen 2002, 66-67; Ruohonen & Salmela 2005, 66-70)

Järjestelmän huolellinen määrittäminen on tärkeää, koska kaikki myöhemmät vaiheet pohjautuvat sille. Muutosten tekeminen muuttuu sitä kalliimmaksi mitä valmiimpi järjestelmä on, joten jo senkin vuoksi on syytä yrittää saada alkuperäiset määrittelyt niin kattaviksi kuin mahdollista, vaikka kaikkia projektin aikana mahdollisesti tai sitten ei ilmeneviä muutostarpeita ei aina voidakaan ennakoita. On myös tärkeää tuottaa riittävästi dokumentaatiota, jotta kaikki järjestelmän parissa nyt ja tulevaisuudessa työskentelevät voivat ymmärtää mitä se tekee ilman tarvetta tutustua sen varsinaiseen ohjelmakoodiin. (Kasurinen 2013, 12; Pohjonen 2002, 29, 32, 79; Ruohonen & Salmela 2005, 66)

Tilaajan prosessien analysointi on tärkeää, jotta voidaan varmistaa, että sekä järjestelmän valmistajat että sen tulevat käyttäjät puhuvat samaa kieltä. Huolellisesti tehtyjä kaavioita voi paikoitellen myös käyttää sinällään esimerkiksi tietokannan suunnittelussa. Niitä käytetään myös järjestelmän testaamisessa, jotta voidaan tarkistaa täyttääkö

lopullinen järjestelmä sille asetetut tavoitteet. Kaikkia kaavioita ei kuitenkaan tehdä kaikissa tilanteissa, vaan ainoastaan silloin kun niille koetaan olevan erityisesti tarvetta, ei-vähän ne synny itsestään vaan edellyttävät aina ylimääräistä työtä sekä silloin kun ne laaditaan, että silloin kun jokin kuvattu asia muuttuu ja sille on siis laadittava uusi kuvaus. (Kasurinen 2013, 29, 43; Pohjonen 2002, 35-36; Ruuhonen & Salmela 2005, 66)

Toiminnan määrittelyn lopputuloksena tulisi olla selvillä kaikki, mitä järjestelmällä voi periaatteessa tehdä. Kaikki mahdolliset käyttötapaukset tulisi kuvata ja niiden toteutus suunnitella. Tämä ei tietenkään tarkoita, että esimerkiksi kirjastojärjestelmässä tulisi kuvata erikseen jokaisen mahdollisen niteen palautus, vaan ainoastaan ne tapaukset jotka käytännössä eroavat toisistaan esimerkiksi osanottajien tai toimenpiteiden osalta. Kun suunnittelu on tehty riittävän huolella, pitäisi toteutuksen, eli varsinaisen ohjelmoinnin, olla hyvin suoraviivaista, onhan kaikki tarpeelliset määrittelyt jo tehty joten ne tarvitsee enää toteuttaa. (Pohjonen 2002, 31-34)

Koska tämä nimenomainen projekti ei edellytä kommunikaatiota eri henkilöiden välillä sen selvittämiseksi, tekeekö järjestelmä oikeita asioita, eikä edes sen vuoksi, että useampi ihminen olisi sitä suunnittelemassa ja toteuttamassa, arvelin sanallisten kuvausten järjestelmän tarkoituksesta ja tehtävistä riittävän tämän vaiheen dokumentaatioksi. Siitä johtuen en perehdy tämän enempää erilaisiin kuvantamistapoihin ja kaaviotyyppeihin. Käytän kuitenkin UML-kaavioita tietokannan kuvausten tekemiseen, joten selostan UML-kaavioiden pääpiirteet tietokannan suunnittelusta kertovassa osassa.

### 3.3 Relaatietietokannan suunnittelu

Sanalla *tietokanta* tarkoitetaan jotakin tarkoitusta palvelevaa kokoelmaa varastoitua tietoa, johon on mahdollista päästä käsiksi ja joka esitetään johdonmukaisesti. Tietokanta saattaa koostua yhtä hyvin fyysisistä papereista kuin sähköisistä dokumenteistakin. Se voi sisältää tekstiä ja numeroita, mutta myös esimerkiksi kuvia, kaavioita ja ääniä. (Page & Hughes 1999, 10)

Oikein suunniteltu tietokanta mahdollistaa tarkkojen ja ajantasaisten tietojen saamisen käyttöön: jos tietokanta antaa virheellistä tietoa, myös näihin tietoihin perustuvat pää-

tökset ovat todennäköisesti vääriä. (Microsoft, Tietokannan suunnittelun perusteet) Verkkoympäristössä virheellisiä tietoja sisältävät tietokannat voivat aiheuttaa muitakin ongelmia, kuten esimerkiksi sen, että käyttäjät eivät pääse kirjautumaan sisään käyttäjätunnuksillaan. Tällainen ongelma voi tietysti johtua myös virheestä ohjelmakoodissa, mutta se ei vähennä tietokannan hyvän suunnittelun tärkeyttä.

Yleisimmillä nykyään käytössä olevilla tietokannan hallintajärjestelmillä hallinnoidaan relaatiotietokantoja. Muunkinlaisia tietokantamalleja on olemassa, mutta niihin ei ole tässä yhteydessä tarkoituksenmukaista perehtyä. Suunniteltavaa järjestelmää on tarkoitus voida käyttää tavallisissa tietokantaympäristöissä, eikä näin ollen ole tarpeen suunnitella sen toimivuutta muulla kuin relaatiotietokantamallilla. Tarkoituksena on, että järjestelmä voidaan asentaa mille tahansa web-palvelimelle, joka mahdollistaa tietokantojen käytön, tai ainakin mille tahansa web-palvelimelle, jolle phpBB-keskustelufoorumikin voidaan asentaa, sillä nämä kaksi määrittystä eivät välttämättä ole täysin yhteenevät: phpBB asettaa tiettyjä reunaehtoja sille, mihin se voidaan asentaa.

### 3.3.1 Relaatiotietokanta lyhyesti

Relaatiotietokantamalli on IBM:n tutkijan E. F. Coddin 1970-luvulla laatima matematiikan menetelmiin perustuva teoria tietokantojen käsittelystä. Relaatiotietokannan tauluilla voi olla suhteita toisiinsa, tämä tosin pätee muissakin tietokantamalleissa. Mallin nimi ei kuitenkaan tule tästä, vaan matematiikan joukkoteorian termistä. (Hernandez 2000, 11-12)

Tiedot talletetaan relaatiotietokantamallissa tietokantataulua vastaavina relaatioina, jotka koostuvat tietueista eli monikoista. Jokainen tietue tunnistetaan siinä olevan kentän, attribuutin, perusteella – jokaisessa tietueessa on oltava yksi kenttä, jonka arvo poikkeaa kaikkien muiden tietueiden vastaavan kentän arvosta. Tietueiden ja kenttien järjestyksellä ei sen sijaan ole mitään merkitystä: tiedot haetaan tietueen kenttien arvon, ei sijainnin, perusteella. Taulujen välillä voi olla suoria ja epäsuoria yhteyksiä, ja tietoja voi hakea mistä kentästä tahansa. (Hernandez 2000, 12-13)

Jotta taulu olisi relaatiotaulu, siinä on kuitenkin oltava tiettyjä erikoisominaisuuksia. Ensinnäkin yhdessä kentässä saa olla vain yksi erottamaton tieto, esimerkiksi joko etunimi tai sukunimi, mutta ei molempia. Yhdessä sarakkeessa saa olla vain yhdentyypistä tietoa, ja jokaisen tietueen tulee olla ainutkertainen eli jokaisessa tietueessa tulee olla vähintään yksi kenttä, jonka arvo poikkeaa muista kentistä. (Page & Hughes 1999, 15-16)

Relaatiotietokannan kanssa kommunikoidaan SQL-kielellä, joka on relaatiotietokantojen hallinnointiin kehitetty kieli. Tällä kielellä kirjoitetaan komentoja, joilla tietokannasta haetaan, lisätään ja poistetaan tietoa. SQL mahdollistaa monimutkaiset hakukyselyt, jotka kohdistuvat useaan tauluun samanaikaisesti, ja monet muut operaatiot, jotka perustuvat matematiikkaan. Relaatiomalli on matematiikkaan perustuva malli, ja sitä hallinnoidaan matemaattisilla operaatioilla. (Page & Hughes 1999, 15-16)

### 3.3.2 Kenttien määrittäminen

Relaatiotietokannan suunnittelu aloitetaan tutustumalla nykyiseen tietokantaan, mikäli sellainen on olemassa, sekä haastatteleamalla tulevia käyttäjiä. Tietokannan ei tarvitse olla sähköinen, se voi olla myös papereihin tai ihmisten muistiin perustuva. Suunnittelua aloitettaessa täytyy selvittää, mitä tietoja organisaatio käyttää ja miten se niitä käyttää, ja paras tapa tämän selvittämiseen on tutkia niitä nykyisestä tietokannasta. Uutta tietokantaa ei kuitenkaan tule rakentaa vanhan kopioksi, sillä uutta tietokantaa ei tarvittaisi, jos vanha vastaisi täysin tarkoitustaan. Vanhasta tietokannasta tulee hyödyntää vain ne osat, jotka ovat hyvin tehtyjä ja tarpeellisia. (Hernandez 2000, 97-99)

Haastateltaessa käyttäjiä web-sisällönhallintajärjestelmän suunnittelua varten on selvitettävä minkälaisia sivutyyppejä sovelluksessa tarvitaan ja minkälaisia tietoja niistä tulee tallentaa. Lisäksi on tarpeellista tietää luodaanko tallennetusta aineistosta automaattisia luetteloita ja käytetäänkö sivuilla suuria kuva- tai videotiedostoja, jotka on suorituskyvyn vuoksi parempi tallentaa tietokannan ulkopuolelle. (Samela 2002, 76)

Analyysin aikana on syytä alkaa tehdä luetteloja niistä tietokentistä, joita tulevaan tietokantaan tarvitaan. Luettelon ei suinkaan tarvitse olla heti täydellinen: sitä tullaan säätä-

mään myöhemmin. Tässä vaiheessa on tärkeää listata kaikki ne asiat, jotka vaikuttavat siltä kuin ne voisivat olla tärkeitä tietokannan kannalta. Sen jälkeen on syytä alkaa laatia prototyyppisiä siitä, millaisia raportteja tietokannasta tulisi tulostaa, ja pohtia, tarvitaanko niiden esittämiseen mahdollisesti joitakin muita tietoja kuin ne, joita tietokentiksi on luonnosteltu: jos esimerkiksi halutaan lähettää sähköposteja asiakkaille, on asiakastietoihin luonnollisesti tallennettava asiakkaan sähköpostiosoite. Jos tällaisia uusia tietoja löydetään, myös ne on syytä sijoittaa alustavaan luetteloon tietokentistä. (Microsoft, Tietokannan suunnittelun perusteet)

Vanhan tietokannan analysoinnin ja tulevien käyttäjien haastattelemisen jälkeen aletaan säätää alustavaa luetteloa tietokentistä. Jos luettelossa mainitaan sama asia kahdesti, selvitetään, onko kyseessä tosiaan sama asia vai samalta vaikuttava asia. Samaa asiaa kahdesti kuvaavat kentät poistetaan, samalta vaikuttavia muokataan siten, että ne kuvaavat paremmin sitä, mitä niillä yritetään kuvata. Jos taas luettelossa on monia nimiä samalle asialle, valitaan näistä nimistä se, joka kuvaa asiaa parhaiten. Lopuksi on varmistettava, että jokainen luettelon kenttä kuvaa yhtä ominaisuutta, eikä esimerkiksi koelmaa, ja että yhtäkään luettelon kentistä ei ole laskettu minkään muun kentän arvoista. (Hernandez 2000, 137-143)

Jokaisella kentällä on myös oltava tietotyyppi, joka määrittää minkä muotoista tietoa siihen tallennetaan. Eri tietokantajärjestelmillä on omat tietotyyppimäärityksensä, joilla kaikilla on omat etunsa ja haittansa. Yleisimpiä ovat eripituiset merkkijonot, eripituiset luvut, suuret tekstit ja binääritietokentät (joihin voi tallentaa esimerkiksi tiedostoja). Tietotyyppiä määrittäessä on otettava huomioon, millaista tietoa kenttä sisältää: puhelinnumerolle ei kannata antaa tietotyyppiä, joka mahdollistaa 5000 merkin syöttämisen. Virheellisesti määritetyt kentät kuluttavat ylimääräistä levytilaa ja kuormittavat järjestelmää turhaan. (Datanamic, Introduction to database design)

### 3.3.3 Taulujen määrittäminen

Kun tarvittavat tiedot on kerätty, suunnitellaan tietokannan taulurakenteet. Aluksi tutkitaan alustavaa kenttien luetteloa, ja mietitään, mitä asiaa mikäkin kenttä kuvaa. Nämä asiat kirjoitetaan alustavaan taululistaan. Kun kenttäluettelo on käyty läpi, tutkitaan

käyttäjien haastatteluissa esittämät mielipiteet siitä, mitä tietoja tietokannan tulisi sisältää, ja luodaan niiden pohjalta toinen taululista. Nämä listat yhdistetään, ja tuloksena syntyvä luettelo tarkastetaan kaksoiskappaleiden varalta. Identtisistä tauluista jätetään vain yksi, osittain samaa asiaa kuvaavat muokataan niin, että jokainen kuvaa selkeästi omaa asiaansa. Sitten sijoitetaan kentät niihin tauluihin, joihin ne sopivat parhaiten. (Hernandez 2000, 151-157, 174)

Kun on valittu, mitä tietoa tauluun tallennetaan, on varmistettava, että kuhunkin tauluun tosiaan tallennetaan vain sellaista tietoa joka kuuluu sinne. Esimerkiksi ”Tuotteet”-tauluun ei kuulu tallentaa tuotteiden toimittajien osoitetietoja, koska toimittajien osoitteet eivät ole tuotteiden vaan niiden toimittajien ominaisuuksia. Tiedoista tulee tarkastaa myös, että yhteen tauluun ei erehdyksessä tallenneta kahden taulun tietoja tilanteessa, jossa toista taulua ei ole suunniteltu: esimerkiksi tilanteessa, jossa tuotteiden toimittajille ei ole suunniteltu omaa tauluaan, voidaan ajatella tallentaa nämä tiedot samaan tauluun tuotteiden kanssa; jos yhdellä toimittajalla on useampi kuin yksi tuote, samat tiedot kuitenkin toistuvat taulussa useamman kerran, mikä on päivityksen kannalta hankalaa, jos toimittajan tiedot muuttuvat. Näin ollen tuotteille ja toimittajille selvästikin tarvitaan eri taulu, ellei sitten olla ehdottoman varmoja ettei missään tilanteessa yhdellä toimittajalla voi olla enempää eikä vähempää kuin yksi tuote, jota siltä tilataan ja joka on merkitty tietokantaan. (Microsoft, Tietokannan suunnittelun perusteet)

Näin luoduista tauluista on tarkastettava, että kaikki kentät ja taulut ovat relaatiomallin mukaisia. On siis selvitettävä, onko kaikkien kenttien sisältö jakamatonta vai sisältääkö jokin kentistä mahdollisesti enemmän kuin yhden tiedon. Jos jossakin kentässä on useampi kuin yksi arvo, on selvitettävä, miten tiedot saadaan purettua. Useimmiten on luotava näille tiedoille uusi taulu. Esimerkiksi taulussa ”tilaukset” oleva sarake ”tilatut tuotteet” on muutettava ”tilaukset”-tauluun viittaavaksi tauluksi. Tätä kutsutaan ensimmäiseksi normaalimuodoksi (1NF). Jotta tietokanta olisi relaatiomallin mukainen, sen on noudatettava normaalimuotoja, siis se on normalisoitava. Seuraavat normaalimuodot liittyvät kuitenkin taulujen pääavaimiin, joten ennen kuin niitä voidaan selvittää, on tauluille määritettävä pääavaimet. (Page & Hughes 1999, 25-26)

Kaikkia tietoja ei kuitenkaan välttämättä tarvitse jakaa – tai paremminkin, tiedot tulee jakaa osiin joista on jotain hyötyä. Vaikka puhelinnumeron teknisesti voisikin jakaa



niin, että jokainen numero olisi tallennettu erikseen, ei tästä käytännössä olisi mitään iloa (paitsi jos ollaan tekemässä tilastoa puhelinnumeroissa käytetyistä numeroista). Sen sijaan henkilön nimen jakaminen etu- ja sukunimeen on hyvinkin hyödyllistä, varsinkin jos tietoa jossakin vaiheessa etsitään näillä kriteereillä. (Microsoft, Tietokannan suunnittelun perusteet)

#### 3.3.4 Pääavaimet

Pääavain eli perusavain on kenttä tai kenttien yhdistelmä, jonka avulla taulun tietueet yksilöidään, esimerkiksi tuotetunnus tai työntekijän tunnusnumero. Jokaisella taululla tulee olla yksi ja vain yksi pääavain. Pääavaimen lisäksi taulussa voi kuitenkin olla avainehdokkaita, jotka ovat pääavaimen vaatimukset täyttäviä kenttiä tai kenttien yhdistelmiä. Pääavaimet valitaan avainehdokkaiden joukosta, mutta taulussa voi olla enemmänkin avainehdokkaita kuin se, joka lopulta valitaan pääavaimeksi. (Hernandez 2000, 212-213; Microsoft, Tietokannan suunnittelun perusteet)

Pääavainten luominen aloitetaan avainehdokkaiden määrittelyllä. Avainehdokkaan tulee olla kenttä, jonka arvo on jokaisessa tietueessa erilainen. Avainehdokkaana voi toimia myös useamman kentän yhdistelmä, ja tällöin riittää, että kenttien yhdistelmän arvo on tietueissa erilainen, yhdistelmän osana olevissa yksittäisissä kentissä voi esiintyä samoja arvoja. Avainehdokka saa kuitenkin sisältää vain niin monta kenttää kuin on välttämätöntä, eli jos kyseessä on kenttien yhdistelmä, kunkin kentän on oltava tarpeen tietueen yksilöimiseksi. (Hernandez 2000, 213)

Avainehdokkaaksi kelpaavan kentän arvo ei saa olla tyhjä, tai jos kyseessä on kenttien yhdistelmä, ainoankaan yhdistelmässä esiintyvän kentän arvo ei saa olla tyhjä. Tästä seuraa, että kentän täyttäminen ei saa olla vapaaehtoista. Avainehdokkaan arvon tulee vielä olla sellainen, että sen arvoa ei tarvitse vaihtaa kuin pakottavasta syystä. (Hernandez 2000, 213-214)

Jos taulusta ei löydy sopivia avainehdokkaita, sellainen voidaan luoda keinotekoisesti. Tällöin tauluun luodaan kenttä, joka täyttää avainehdokkaan ehdot aina ja joka tilanteessa, yleensä jonkinlainen juokseva numerokenttä. Saattaa nimittäin olla, että jokin

kenttä näyttää sopivalta avainehdokkaalta, mutta tulevaisuudessa siihen onkin tarvetta sijoittaa arvoja, jotka eivät poikkeakaan toisistaan. Jos kuitenkin tiedetään, että valittu avainehdokas on tulevaisuudessakin pätevä avainehdokas, keinotekoiselle avainehdokkaalle ei tietenkään ole tarvetta. (Hernandez 2000, 218-220)

Kun kaikki taulun avainehdokkaat on määritelty, niistä valitaan se, joka parhaiten soveltuu sille pääavaimeksi. Pääavain on se avain, jolla taulun tietueet tunnistetaan ja siten erotetaan toisistaan. Näitä avaimia tulee jokaisessa taulussa olla yksi ja vain yksi, joko yksi kenttä tai yksi kenttien yhdistelmä. Tähän tarkoitukseen tulee valikoida se avainehdokas, jolla järjestelmän tulevat käyttäjät mieluiten tietueet tunnistavat, tai se avainehdokas, jossa on vähiten kenttiä. Ennen pääavaimen lopullista valintaa on vielä tarkistettava, että se viittaa jokaiseen tietueen kenttään suoraan eikä välillisesti, jonkin muun kentän arvon kautta. Tätä kutsutaan toiseksi normaalimuodoksi (2NF). Kun jokaisen kentän arvo riippuu pääavaimesta ja vain pääavaimesta, ei minkään muun kentän arvosta, puhutaan kolmannelle normaalimuodosta (3NF). Lopuksi on vielä tarkastettava, että pääavaimen arvo ei riipu minkään muun sarakkeen arvosta. Tätä kutsutaan Boyce Codd -normaalimuodoksi (BCNF). Normaalimuotoja on enemmänkin, mutta tietokanta on sujuvan käytön kannalta riittävän normalisoitu, kun se noudattaa BCNF-muotoa. (Hernandez 2000, 220-224; Page & Hughes 1999, 26-28)

Koska jokaisen kentän sisällön tulee olla riippuvainen vain ja ainoastaan pääavaimesta, ei tietokantatauluun tulisi sisällyttää laskettuja kenttiä, eli kenttiä joiden arvo on laskettu joistakin muista kentistä. Tällaiset tiedot voidaan laskea silloin, kun raporttia näytetään, niitä ei ole tarpeen tallentaa tietokantaan – paitsi tilanteessa, jossa ne eivät olekaan samoja kaikille. Jos myyntiin liittyykin tietty alennusprosentti, joka on myyntikohtainen, ei myynnin loppusummaa voikaan laskea tuotteiden hinnoista. Sama pätee, jos tuotteiden hinnat muuttuvat: ennen hintamuutosta tehdyn myynnin loppusumma ei muutu mainitun muutoksen takia. (Datanamic, Introduction to database design; Microsoft, Tietokannan suunnittelun perusteet)

### 3.3.5 Viiteavaimet ja yhteydet

Tietokannan taulujen välille voidaan muodostaa yhteyksiä, jotka ovat liitoksia tietokannan taulujen välillä. Taulut voidaan liittää yhteen pääavaimen ja viiteavaimen tai sitten erityisen linkitystaulun välillä, joskin linkitystaulussakin käytetään taulujen pääavaimiin viittaavia viiteavaimia. Yhteyksien avulla liitetään kaksi loogisesti toisiinsa liittyvää taulua yhteen, helpotetaan tiedon hakemista monesta taulusta samanaikaisesti ja vähennetään saman asian kirjaamista tauluihin useaan kertaan, mikä helpottaa tiedon pitämistä oikeellisenä. (Hernandez 2000, 273-274; Page & Hughes 1999, 24)

Taulujen väliset yhteydet jaetaan kolmeen eri tyyppiin: yhdestä yhteen, yhdestä moneen ja monesta moneen -yhteyksiin. Yhdestä yhteen -yhteyksissä taulun A yksittäinen tietue voi liittyä yhteen taulun B tietueeseen, ja taulun B yksittäinen tietue voi liittyä yhteen taulun A tietueeseen. Yhdestä moneen -yhteyksissä, relaatiotietokannoissa yleisimmin käytössä olevassa yhteystyypissä, taulun A yksittäinen tietue voi liittyä yhteen tai useampaan taulun B tietueeseen, kun taulun B yksittäinen tietue voi liittyä vain yhteen taulun A tietueeseen. Nämä yhteydet luodaan lisäämällä toiseen tauluista viiteavain, joka viittaa ensimmäisen taulun pääavaimen. Viiteavaimen arvon tulee siis olla jokin viitattavan pääavaimen arvoista, ja kentän tulee muutenkin olla pääavaimen kenttää vastaava. Jos pääavain on kenttien yhdistelmä, viiteavaimenkin tulee olla vastaava kenttien yhdistelmä. (Hernandez 2000, 276-278, 289, 299; Page & Hughes 1999, 20-21)

Monesta moneen -yhteyksissä taulun A yksittäinen tietue voi liittyä yhteen tai useampaan taulun B tietueeseen, ja taulun B yksittäinen tietue voi liittyä yhteen tai useampaan taulun A tietueeseen. Monesta moneen -yhteyksien käyttämiseen liittyy kuitenkin relaatiotietokannan hallintajärjestelmien toteutuksissa ongelmia, joiden vuoksi näitä yhteyksiä ei voida käyttää sellaisinaan. Monesta moneen -yhteyden toteuttaminen vaatii linkitystaulun luomisen tietokantaan. Tällaisessa taulussa on pääavaimena yhteyteen kuuluvien taulujen kenttien pääavainten muodostama yhdistelmä, joiden lisäksi tauluun kuuluvat mahdolliset yhteyteen liittyviä ominaisuuksia kuvaavat kentät. (Hernandez 2000, 279, 293-297; Page & Hughes 1999, 22)

Taulujen välisten yhteyksien luominen on syytä aloittaa jälleen haastattelemalla niitä, jotka tuntevat organisaation toiminnan parhaiten. Näillä ihmisillä on todennäköisesti

myös paras käsitys siitä, miten tietokantaan talletettavat tiedot liittyvät toisiinsa. Osa yhteyksistä on välttämätöntä luoda tietokannan suunnittelijan näkemyksen perusteella, sillä osa yhteyksistä liittyy tietokannan tekniseen toteutukseen, joista käyttäjillä ei ole tietoa. Osa yhteyksistä on kuitenkin luotava sen perusteella, mikä suhde tiedoilla on käyttäjien näkökulmasta toisiinsa. (Hernandez 2000, 284-286, 298)

Kun organisaation edustajien haastattelut ja niiden perusteella luotavat yhteydet on tehty, rakenteet tarkastetaan vielä kerran ja mahdolliset virheet korjataan. Sitten yhteyksille asetetaan ominaisuudet, jotka määrittävät sen, mitä tapahtuu jos yhteyteen kuuluvia tietueita yritetään poistaa tai niiden pääavaimia muokata. Mahdollisuuksia ovat muutosten ja poistojen estäminen, muutosten suorittaminen kaikissa muutettavaan tietueeseen liittyvissä tietueissa, ja kaikkien poistettavaan tietueeseen liittyvien tietueiden poistaminen. (Hernandez 2000, 298-299, 301-302)

### 3.3.6 Denormalisointi

Tietokanta on normalisoitava, jotta tietojen pitäminen ehyenä on helppoa ja jotta tallennustilaa tietokannassa säästetään (ylimääräiset moninkertaiset tiedot tietokannassa kulluttavat levytilaa turhan takia). On kuitenkin tilanteita, joissa normalisoituun tietokantarakenteeseen voi olla tarpeellista palauttaa normalisoinnin vastaisia rakenteita. Tätä kutsutaan denormalisoinniksi. (Microsoft, Lesson 3: Optimizing the Database Design by Denormalizing)

Tarve denormalisoinnille voi syntyä siksi, että normalisoitu tietokanta näyttää vain sen hetkisen tilanteen, ja tiedoista halutaan tallentaa myös jonkinlainen historia, koska ne voivat muuttua matkan varrella (esimerkiksi tuotteiden hinnat; hinnanmuutoksen ei pitäisi vaikuttaa ennen muutosta tehtyjen ostojen loppusummaan). Tällä tavalla merkittyihin tietoihin liittyy aina jollakin tapaa ajankohta, minkä vuoksi tietojen eheyden riskeeraaminen on perusteltua: tietokannan tietoja käytetään kuvaamaan tilanteita eri ajanhetkinä. (Hernandez 2000, 410; Microsoft, Lesson 3: Optimizing the Database Design by Denormalizing)

Toinen syy normalisointisääntöjen rikkomiselle voi olla tietokannan suorituskyky: koska tietokantakyselyn tekeminen siten, että siihen yhdistetään tietoja kahdesta eri tietokantataulusta, kuormittaa järjestelmää enemmän kuin kyselyn tekeminen yhdestä taulusta, voi olla perusteltua lisätä ne tiedot, joita useimmiten tarvitaan kahdesta eri taulusta kyselyitä tehtäessä, toiseen tauluista – mieluiten siihen, josta suurin osa tiedoista haetaan. Näin rikotaan sääntöä, jonka mukaan toistuvat esiintymät samasta tiedosta ovat pahasta, mutta mahdollisesti parannetaan järjestelmän suorituskykyä, erityisesti jos kyseisiä hakuja tehdään paljon. (Microsoft, Lesson 3: Optimizing the Database Design by Denormalizing)

Tietokannan suorituskykyyn liittyy myös tilanne, jossa tietokantaan päätetään sittenkin sijoittaa laskettu kenttä. Esimerkiksi varastokirjanpidossa on tehokkaampaa tallentaa tietokantaan tieto varaston senhetkisestä tilanteesta kuin suorittaa hakukyselyt kaikista varastoon kohdistuneista ostoista ja myynneistä sen tarkistamiseen, kuinka monta kappaletta jotakin tuotetta on varastossa juuri sillä hetkellä: hakukyselyihin kuluu aina aikaa, ja mitä suurempi kysely, sitä enemmän aikaa ja tietokoneiden laskentakapasiteettia siihen kuluu. (Microsoft, Lesson 3: Optimizing the Database Design by Denormalizing)

Suorituskykyä parantamaan tarkoitettua denormalisointia tulee kuitenkin käyttää vasta viimeisenä keinona, ja sen käyttöä tulee harkita erityisen tarkasti punniten haittoja ja hyötyjä. Se saattaa aiheuttaa suuriakin ongelmia, ellei järjestelmä osakaan huolehtia tietojen eheydestä. Koska moninkertaisten tietojen tallentaminen vaikeuttaa niiden ylläpitämistä, on selvää, että tämänkaltaisen denormalisointi saattaa aiheuttaa huomattavia ongelmia tietojen oikeellisuudelle. Ennen tietojen tallentamista moninkertaisina tulee tarkistaa voisiko järjestelmän suorituskykyä parantaa jollakin muulla tavalla, esimerkiksi laitteistoa päivittämällä. On myös mahdollista, että tietokannan rakennetta tai tietokantaa käsittelevän ohjelmakoodin toimintaa pitäisi parantaa. Vasta, jos nämä ratkaisut osoittautuvat toimimattomiksi tai riittämättömiksi, tulee tietokannan rakennetta tietoisesti suorituskyvyn nimissä huonontaa, sillä sitä denormalisointi käytännössä tarkoittaa. (Hernandez 2000, 410-413; Microsoft, Lesson 3: Optimizing the Database Design by Denormalizing)

Jos normalisointisääntöjä päädytään rikkomään, on oltava tarkkana siitä, että jossakin kohtaa järjestelmässä pidetään huolta tietojen eheydestä. Ellei siitä huolehdi tietokanta,

on asia hoidettava ohjelmallisesti. Erityisen tarkkana on oltava laskettujen kenttien suhteen: mikäli tietokantaan halutaan tallentaa tieto esimerkiksi jonkin tuotteen senhetkisestä kappalemäärästä, on myös pidettävä huolta siitä että kyseistä arvoa muutetaan aina kun tuotteiden kappalemäärä todellisuudessa muuttuu. (Hernandez 2000, 411; Microsoft, Lesson 3: Optimizing the Database Design by Denormalizing)

Denormalisointi on aina suunniteltava ja dokumentoitava tarkasti, jotta tietokantaa käyttävän järjestelmän ohjelmoijat osaavat sijoittaa tarvittavan tietojen päivityslogiikan ohjelmakoodiin ja jotta muutokset tietokantaan voidaan kumota, mikäli ne eivät olekaan saaneet aikaan toivottuja hyötyjä tai ovat aiheuttaneet epätoivottuja ongelmia. Ilman huolellista suunnittelua ja toteutusta normalisointisääntöjen rikkominen aiheuttaa juuri niitä vaikeuksia, joita normalisoinnilla pyritään ehkäisemään – eihän koko sääntöjä muuten edes tarvittaisi. Siihen ei koskaan tule turvautua ilman perinpohjaista harkintaa, eikä missään tapauksessa ennen kuin tietokanta on laadittu normaalisääntöjen mukaisesti. (Hernandez 2000, 410, 413-414; Microsoft, Lesson 3: Optimizing the Database Design by Denormalizing)

### 3.3.7 UML tietokannan kuvauksessa

UML, lyhenne sanoista Unified Modeling Language, on kieli, joka on kehitetty kuvaamaan ohjelmistoja ja niiden toimintaympäristöjä tavalla, joka sekä auttaa järjestelmän toteuttajia työssään että on ymmärrettävissä myös niiden näkökulmasta, jotka eivät ymmärrä mitään varsinaisesta ohjelmointityöstä. Se julkaistiin alkujaan vuonna 1997, vaikka sen kaaviotyyppejä oli käytetty jo aiemmin. UML:n tarkoituksena on tarjota yhteinen kuvantamistapa kaikille määrittely- ja suunnitteluvaiheissa tehtäville kaavioille ja esittää ohjelma kokonaisuutena samaan tapaan kuin rakennuspiirustukset esittelevät talon. (Kasurinen 2013, 29-30)

Koska UML-kieli on tarkoitettu kuvaamaan monia eri asioita järjestelmästä, siihen kuuluu useita erilaisia kaavioita: käyttötapauskaavioita käyttötapausten kuvaamiseen, luokkakaavioita ohjelmiston käyttämien kohteiden määrittämiseen, oliokaavioita kuvaamaan järjestelmää satunnaisella ajanhetkellä, komponenttikaavioita järjestelmän eri osien sijoittumisen ja yhteyksien kuvaamiseen, toteutuskaavioita järjestelmän fyysisen raken-

teen määrittämiseen, toimintokaavioita järjestelmän toimintalogiikan kuvaamiseen, pakkikaavioita järjestelmän eri osien välisten yhteyksien kuvaamiseen, rakennekaavioita järjestelmän yksittäisten osien rakenteen kuvaamiseen, tilakaavioita järjestelmän toimintatilojen mallintamiseen, kommunikaatiokaavioita ja sekvenssikaavioita järjestelmän eri osien välisen viestinvaihdon kuvaamiseen, ajoituskaavioita eri signaalien ajoituksen määrittämiseen sekä kokoavia vuorovaikutuskaavioita, jotka määrittävät miten muut kaaviot liittyvät toisiinsa. (Kasurinen 2013, 30-41) Tässä työssä käytetään luokkakaavioita tietokannan määrittämiseen, mutta ei muita kaaviotyyppisiä, joten esittelen niitä hieman tarkemmin siinä määrin kuin se on työn kannalta relevanttia.

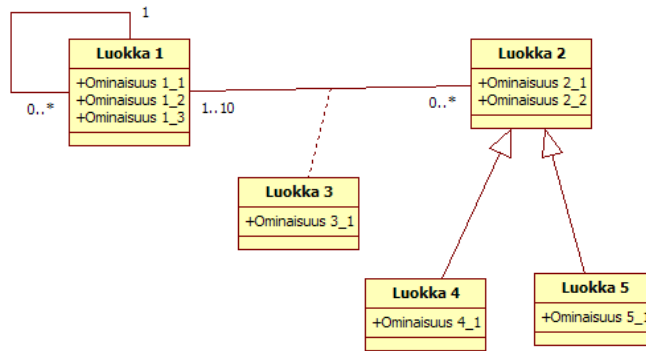
Luokkakaaviossa kuvataan sovelluksen luokat (tässä tapauksessa tietokantataulut), niiden sisältö (tässä tapauksessa tietokantataulujen kentät) ja niiden väliset suhteet. Luokka kuvataan kolmiosaisena laatikkona, jonka yläosassa on luokan nimi, keskiosassa ominaisuudet (kentät), ja alaosassa operaatiot (tietokantataulun ollessa kyseessä operaatioita ei ole, mutta esimerkiksi ohjelmakoodin tai todellisen elämän tilanteen mallintamisessa niitä saatetaan tarvita). Ominaisuuksista voidaan kuvata paitsi nimi myös tietotyyppi ja oletusarvo, ja operaatioista mitä parametreja se tarvitsee ja minkä tyyppisen arvon se palauttaa, jos kyse on ohjelmakoodin kuvauksesta. (Pohjonen 2002, 153-154)

Luokkien väliset yhteydet kuvataan viivoilla, joiden päät ovat niissä luokissa, joiden välillä yhteys on. Yhteyteen kuuluu myös lukuarvo: kuvaus siitä, kuinka monta luokan edustajaa (tässä tapauksessa tietuetta) voi liittyä yhteen yhteyden ilmentymään (tässä tapauksessa tietueeseen yhteyden toisessa päässä olevassa tietokantataulussa). Esimerkiksi merkintä ”1..10” tarkoittaa, että yksittäisessä yhteydessä tulee olla vähintään 1 ja enintään 10 luokan ilmentymää, ”0..\*” tarkoittaa, että luokan ilmentymiä ei välttämättä tarvitse olla yhtään, mutta niitä voi olla kuinka paljon tahansa. Yhteyksillä voi olla myös omia ominaisuuksia, kuten luokilla. Nämä kuvataan luokkana, joka yhdistetään yhteyteen katkoviivalla. (Pohjonen 2002, 154-155)

Luokkien välisiin yhteystyyppisiin kuuluu vielä perintäsuhde, johon kuuluu yksi yläluokka ja kaksi tai useampia alaluokkia. Tällainen suhde voi tulla kyseeseen, jos luokkien välillä on osittaisia yhtäläisyyksiä: yhteiset piirteet kirjataan yläluokkaan, josta alaluokat ”perivät” ne. Perintä kuvataan kolmiolla, josta on yhteys ylä- ja alaluokkiin. Ylä-

luokka on yleistys luokan yleisistä piirteistä, ja alaluokat ovat yläluokan erikoistapauksia. (Pohjonen 2002, 155-156, 166-167)

Luokkakaavion pääpiirteet on esitetty kuvassa 1.



Kuva 1. Luokkakaavion pääpiirteet. Luokalla 1 on suhde itseensä, samoin luokkaan 2. Luokassa 3 on luokkien 1 ja 2 välisen yhteyden ominaisuudet. Luokka 2 on luokkien 4 ja 5 yläluokka, josta mainitut luokat 4 ja 5 periytyvät.

UML-kaaviot on oikeastaan tarkoitettu ohjelmistojen eikä tietokantojen kuvaamiseen (Wiio 2004, 107), mutta koska tietokoneelleni sattui olemaan asennettuna ohjelma, jolla kyseisiä kaavioita voi tehdä, päätin käyttää sitä tietokannan mallintamiseen, sillä se oli helpompaa kuin piirtää kaikki taulut ja niiden väliset yhteydet käsipelillä. En tosin ole erityisen hyvä käyttämään kyseistä ohjelmaa, en ole edes varma voiko sillä tehdä tiettyjä asioita oikein, joten kaavioissa voi hyvinkin olla erilaisia teknisiä puutteita viralliseen UML-määrittelykseen verrattuna. Luullakseni ne kuitenkin ovat riittävän selkeitä välittämään sen tiedon, mitä varten ne on tehty, eikä eroavuuksia standardiin näin ollen varmaankaan tarvitse ottaa liian vakavasti.

### 3.4 Web-käyttöliittymän suunnittelu

Käyttöliittymä tarkoittaa tapaa, jolla järjestelmä kommunikoi käyttäjän kanssa, ja sellaisena se on koko järjestelmän tärkein osa. Hyvän käyttöliittymän peruseriaatteet ovat johdonmukaisuus (sama komento tarkoittaa samaa asiaa kaikkialla järjestelmässä), selkeys (kokonaisuus on hahmotettavissa eikä ole sekava), helppous (käyttöliittymä on yksinkertainen omaksua), turvallisuus (mitään pahaa ei pääse tapahtumaan vahingossa)



ja taloudellisuus (vakiokäyttäjän ei tarvitse tehdä kaikkea pitkän kaavan mukaan). Käyttöliittymän tulee keskustella käyttäjän kanssa käyttäjän termien ja tarpeiden mukaisesti, ei varsinaisen toteutuksen teknisillä termeillä. Kirjanpitäjä ei tee mitään sillä tiedolla, että kirjanpito-ohjelman tiedot tallennetaan tietokantaan tietueina. (Ruohonen & Salmela 2005, 74; Wiio 2004, 74-76, 107)

Tässä projektissa suunniteltavaa järjestelmää tullaan käyttämään selaimen avulla web-käyttöliittymän kautta, joten käyttöliittymän suunnittelussa on syytä ajatella juuri web-käyttöliittymää. Tällaiset käyttöliittymät poikkeavat joiltain osin itsenäisten, ilman selainta käytettävien tietokoneohjelmien käyttöliittymistä, mutta koska järjestelmää ei ole tarkoitus rakentaa käytettäväksi muussa kuin web-ympäristössä, tässä ei ole tarpeen selvittää eroavaisuuksia web- ja muiden käyttöliittymien välillä.

Web-käyttöliittymän suunnitteluun on olemassa kaksi eri näkökulmaa. Toinen korostaa taiteellisuutta, toinen käytännöllisyyttä. Käytettävyys on webissä kuitenkin tärkeintä, sillä jos sivuston käyttö on vaikeaa, sitä ei mielellään käytetä. (Nielsen 2000, 9-14)

Tärkein tapa saada web-sivustosta helppokäyttöinen on rakentaa se niin selkeäksi kuin mahdollista. Sivuston ei pitäisi tarvita ylimääräisiä selityksiä toimiakseen. Toisaalta sivustot, joiden ensisijaisena tarkoituksena on palvella kokeneita käyttäjiä, voivat vaatia käyttäjiltään hieman enemmän opettelua. Siitä huolimatta suunnitteluun ja käytettävyyteen kannattaa panostaa, sillä helppokäyttöisen sivuston käyttö on tehokkaampaa kuin vaikeakäyttöisen. (Krug 2006, 11; Nielsen 2000, 274-275)

#### 3.4.1 Käyttöliittymäsuunnittelun yleiset piirteet

Koska käyttöliittymän tarkoituksena on tarjota käyttäjälle välineet tehdä se, mihin käyttäjän on järjestelmää tarkoitus käyttää, on ensimmäinen askel käyttäjäystävällisen järjestelmän ja sen käyttöliittymän suunnittelussa tunnistaa, mitä tarpeita käyttäjällä on (Wiio 2004, 98). Tietysti voidaan suunnitella myös mahdollisimman vaikeasti käytettävä, käyttäjäepäystävällinen järjestelmä, mutta sellaisen tekemisessä ehdoin tahdoin ei ole paljoakaan järkeä. Tarpeiden kartoittamisessa on otettava huomioon myös käyttäjän näkökulma siitä, mitä hän tarkkaan ottaen on tekemässä ja miksi (Wiio 2004, 83-84).

Järjestelmän toimintaympäristöä mallinnettaessa syntyneitä käyttötapausselvityksiä tarvitaan myös käyttäjäystävällisen käyttöliittymän suunnittelussa. On kuitenkin muistettava, että eri käyttäjillä voi olla toisistaan poikkeavia tavoitteita. Yksi käyttäjä voi esimerkiksi etsiä mahdollisimman halpaa lentoa lomakohteeseensa, toinen etsiä samaiseen kohteeseen lentoa jollekin tietylle päivälle liiketapaamista varten. Kumpikin on tilaamassa lentolippua, mutta kaipaa käyttöliittymältä eri asioita. (Wiio 2004, 98-100)

Käyttöliittymää suunniteltaessa on toki muistettava muutakin. Ei riitä, että ominaisuudet ovat olemassa, niiden pitää olla näkyvillä siten että käyttäjä löytää ne. Laajemmissa ohjelmistoissa ei tietenkään voi kaikkea asettaa samanaikaisesti näytölle, mutta asiat on pyrittävä sijoittelemaan niin, että ne muodostavat loogisen kokonaisuuden ja ovat myös visuaalisesti liitettävissä niihin asioihin, joihin ne loogisesti liittyvät. Toimintoja ei pidä piilottaa niin, että käyttäjä voi löytää ne vain sattumalta ellei ole perinpohjaisesti opiskellut ohjelman käyttöopasta. (Wiio 2004, 125, 129-131, 191)

Vakiintuneita käytäntöjä on myös syytä noudattaa. Esimerkiksi OK- ja Peruuta-painikkeet ovat aina tietyssä järjestyksessä, eikä tämän järjestyksen rikkominen ole suositeltavaa, koska se saisi aikaan väärin nappien painamista osan käyttäjistä klikkaillessa vanhasta tottumuksesta siihen kohtaan, missä napit tavanomaisesti ovat. Järjestelmän sisäisestä yhteneväisyydestä on muutenkin huolehdittava: jos komentopainike saa aikaan jotakin yhdessä osassa järjestelmää, ja tismalleen samalta näyttävä komentopainike saakin järjestelmän toisessa osassa aikaan jotain aivan muuta, käyttäjä hämmentyy. (Wiio 2004, 159-161, 169)

Lopuksi tulee kiinnittää huomiota vielä käyttöliittymän visuaaliseen miellyttävyyteen. Kirjasinkoon, rivivälin ja käytettyjen kirjasinlajien tulee olla sellaisia, ettei tekstin lukemiseen tarvita kohtuuttomia ponnistuksia. Visuaalisen hierarkian tulee vastata loogista hierarkiaa, niin että käyttäjälle ei jää epäselvyyttä siitä, mihin mitkäkin asiat kuuluvat. Värimaailman tulee olla harmoninen niin, ettei se häiritse järjestelmän varsinaista käyttöä vaan tukee käyttöliittymän rakennetta. (Krug 2006, 36-37; Martin, *Effective Visual Communication for Graphical User Interfaces*; Wiio 2004, 203-210)

### 3.4.2 Sivuston latautumisnopeus

Web-sivujen tulee latautua käyttäjälle mahdollisimman nopeasti, jotta tuntuma toiminnan hallitsemisesta säilyy. Vielä tärkeämpää on kuitenkin se, että käyttäjä tietää kuinka kauan latautumista on odotettava. Käyttäjä pettyy, jos nopeaksi odotettu toiminto onkin hidas. Paljon parempi on tilanne, jossa käyttäjä tietää joutuvansa odottamaan kauan. (Nielsen 2000, 43-44)

Nopeasti latautuvat sivut ovat käytännöllisiä myös sivutilan maksajan kannalta. Kevyet, nopeasti latautuvat sivut ovat nimittäin tiedostokooltaan pienempiä kuin raskaat, hitaasti latautuvat. Tiedostokooltaan suuren sivun siirtäminen taas vaatii palvelimelta enemmän tiedonsiirtokaistaa kuin pienen, eikä tiedonsiirto ole ilmaista. Kevyt sivusto on siis taloudellisempi kuin raskas. (Veen 2002, 176)

Paras tapa nopean sivuston tekemiseen on käyttää CSS-tyylitiedostoja, kirjoittaa mahdollisimman vähän HTML-koodausta ja käyttää mahdollisimman vähän grafiikkaa. Sivut voi saada näyttäväksi käyttämällä esimerkiksi väritettyjä taulukkosoluja. Kuvat ja multimediasisällöt on parasta jättää tilanteisiin, joissa niitä todella tarvitaan. Edes sivustolla eteneminen yrityksen ja erehdyksen kautta ei haittaa niin paljoa, jos sivut latautuvat nopeasti. (Krug 2006, 25; Nielsen 2000, 46; Veen 2002, 177-178)

### 3.4.3 Käytön ohjeistaminen

Käyttöliittymän tulisi olla niin helppo, että erillisiä ohjeita käyttöön ei tarvita, sillä ohjeistuksia ei lueta vapaaehtoisesti. Parempi kuin kirjoittaa satojen sivujen käyttöoppaita on tehdä toiminnallisuus niin ilmeiseksi, että ohjeita ei tarvita. Jos ohjeita ei voida kuitenkaan kokonaan poistaa, ne on ainakin kirjoitettava niin ytimekkäiksi kuin mahdollista. (Krug 2006, 47; Nielsen 2000, 129-131)

Ekstranet-tyyppisten järjestelmien käyttäjät saattavat kuitenkin olla motivoituneempia näkemään hieman vaivaa käytön oppimisessa, koska he joutuvat jatkuvasti tekemisiin samojen sivujen kanssa ja tahtovat mielellään käyttää järjestelmää tehokkaammin. Täl-

löin ohjeisiinkin voidaan tutustua, mutta yleensä niihin silti perehdytään vasta, kun esille tullut ongelma ei oman ajattelun avulla ratkea. (Nielsen 2000, 129-131)

#### 3.4.4 Intranet- ja ekstranet-tyyppiset järjestelmät

Intranet on organisaation, yrityksen tai oppilaitoksen sisäisessä käytössä oleva järjestelmä, johon voi olla yhteydessä vain kyseisen organisaation, yrityksen tai oppilaitoksen sisäisessä verkossa. Ekstranet on muutoin vastaava järjestelmä, mutta siihen voi olla yhteydessä myös sisäisen verkon ulkopuolelta, Internetin kautta. Koska ekstranet on kuitenkin yhtä lailla tarkoitettu vain tiettyjen henkilöiden käyttöön, tiedot on useimmiten suojattu jonkinlaisella menetelmällä, joka edellyttää sisäänkirjautumista tai muuta vastaavaa käyttäjätunnistusmenetelmää. (Techterms.com, Extranet; Techterms.com, Intranet)

Intranet- ja ekstranet-tyyppisissä, käyttäjiltään julkisten www-sivujen selaajia läheisempää suhdetta sivuston taustalla olevaan organisaatioon edellyttävissä järjestelmissä voidaan olettaa, että käyttäjällä on motivaatiota opetella päivänselvää monimutkaisempikin käyttöliittymä. Koska käyttäjillä on tavoitteita, joiden toteuttamiseen järjestelmää käytetään, heillä on perusselaajaa enemmän tahtoa oppia käyttämään sivustoa. Lisäksi tällaisten järjestelmien käyttäjillä on yleensä jo perustietoa niistä tehtävistä, joita he ovat suorittamassa. (Nielsen 2000, 265-270)

Ekstranet-tyyppisen järjestelmän käytettävyydessä tärkeimmät tavoitteet ovat toiminnan tehokkuus, virheiden minimointi ja muistamisen helppous. Järjestelmän käyttäjän aikaa ei kannata haaskata, koska haaskattu aika olisi ollut mahdollista käyttää johonkin hyödyllisempään ja tuottavampaan. (Nielsen 2000, 274)

#### 3.5 Käytettävyydestaus

Käytettävyydestaus tarkoitusena on tarkistaa, että käyttöliittymä vastaa tarkoitustaan, ja löytää toteutuksesta korjaamista edellyttäviä virheitä. Suunnittelijan tai ohjelmoijan itsensä toteuttama testaus ei ole riittävää, koska järjestelmän parissa pitkään työskennelleet tuntevat sen perinpohjaisesti eivätkä osaa katsella sitä samalla tavoin

kuin sellaiset, jotka eivät etukäteen tunne toteutusta. Tällöin virheet ja toimimattomuudet jäävät helposti huomaamatta, koska testaaja tietää, miten toimintoja on tarkoitus käyttää. Järjestelmän suunnittelijat myös etsivät ulkopuolisia testaa- jia todennäköisem- min vahvistusta omille ajatuksilleen suoranaisten ongelmien sijaan. (Krug 2006, 133-135; Wiio 2004, 223)

Käytettävyydestäukseen osallistuvien henkilöiden tulisi edustaa niitä ryhmiä, jotka tu- levat lopputuotetta käyttämään, ja testauksessa suoritettujen tehtävien tulisi olla sellai- sia, joita järjestelmällä tullaan tekemään. Jos sopivia koehenkilöitä ei kuitenkaan ole saatavilla, testausta voidaan suorittaa myös sellaisilla ihmisillä, jotka eivät missään ta- pauksessa kuulu tulevan järjestelmän käyttäjiin. (Krug 2006, 134; Nielsen 2000, 290)

Käytettävyydestäusta voidaan tehdä sekä laboratoriossa että kenttäolosuhteissa. Labo- ratoriotestaus suoritetaan erillisessä tilassa, jossa koehenkilölle annetaan tehtäviä suori- tettavaksi ja tarkkaillaan hänen toimiaan. Kenttätutkimus toteutetaan tarkkailemalla sivusta koehenkilön työskentelyä tämän suorittaessa normaaleja tehtäviään. (Krug 2006, 139-140; Nielsen 2000, 293)

Käytettävyydestäusta voidaan tehdä myös etätestauksena siten, että testaaja ja koehen- kilö eivät ole samassa paikassa vaan mahdollisesti jopa eri mantereilla. Jos mahdollista, on parempi tehdä testaus jossa koehenkilön kanssa voidaan olla samassa tilassa, mutta jos vaihtoehtona on testaamatta jättäminen, on etätestaus parempi valinta. Etätestauk- sessa koehenkilö käyttää omaa laitteistoaan, ja testaaja joko tarkkailee häntä jonkinlai- sen ohjelmiston välityksellä (valvottu etätestaus) tai vastaanottaa koehenkilön raportin testauksen päätyttyä (valvomaton etätestaus). Erityisesti valvomattomassa etätestauk- sessa on tärkeää kirjoittaa kaikki koehenkilölle annettavat ohjeet huolellisesti, jotta va- raa väärinymmärryksille ei jää: ellei koehenkilö ymmärrä jotain, kyseinen tehtävä jää tekemättä, ja vaikka jonkin toiminnon vaikeus onkin testauksen kannalta merkittävä tie- to, koko testauksen tulos kärsii ellei sitä saada vietyä kokonaan läpi. (Schade 2013)

Web-sivustoa voidaan testata joko testaamalla ymmärtääkö koehenkilö sivuston tarkoi- tuksen tai testaamalla kuinka koehenkilö selviytyy annetun tehtävän suorittamisesta. Testausta voidaan tehdä jo pelkillä luonnoksilla, ja tämä onkin erittäin hyödyllistä, sillä

koehenkilöt uskaltavat kommentoida vapaammin, jos tuote näyttää selvästi keskeneräiseltä. (Krug 2006, 144-145)

Testaaminen tulisi aloittaa heti kun vain on jotakin testattavaa, ja toistaa sitä jokaisessa kehitystyön vaiheessa. Oikeastaan testaus on ajankohtaista jo ennen kuin suunnittelua edes aloitetaan, sillä ennen www-sivujen kehittämisen aloittamista tulee tutustua muihin samankaltaisiin sivustoihin ja testata niitä. Vertailusivustoista voi sitten päätellä mikä toimii ja mikä ei, jotta on helpompi välttää mahdollisia ongelmia ja hyödyntää valmiita hyväksi havaittuja ratkaisuja. (Krug 2006, 144)

Testaus on, kuten kaikki muukin tietojärjestelmän suunnittelussa, dokumentoitava. Dokumentaation tulisi sisältää ainakin testaussuunnitelma, josta käy ilmi mitä testataan, milloin, miten ja miksi. Lisäksi dokumentoituna pitäisi olla ainakin testitapaukset, joilla testaus suoritetaan. Testauksen suorittamisen jälkeen pitää luonnollisesti kirjata testauksen tulokset (ellei sitä tehdä jo testin aikana) ja analysoida ne. (Kasurinen 2013, 104-106; Wiio 2004, 222-223) Tässä työssä ei kuitenkaan päästä itse testaukseen asti, joten tulokset ja niiden analyysi puuttuvat.

#### 4 SUUNNITTELUN KULKU

Aloitin työn tutkimalla sivustoja, joiden kaltaisia järjestelmällä olisi tarkoitus pystyä hallinnoimaan, ja yritin saada selville minkälaisia toistuvia rakenteita niissä esiintyi. Helpommin sanottu kuin tehty, sillä varsinaista yhteistä nimittäjää ei tuntunut löytyvän. Päädyinkin siihen lopputulokseen, että järjestelmän tulee olla mahdollisimman joustava, jotta sillä pystyy hallinnoimaan mahdollisimman monenlaisia sivustoja. Tästä ilmeni suunnittelun edetessä se seuraus, että järjestelmää voi potentiaalisesti käyttää myös muuhun kuin siihen, mihin se on suunniteltu, mikä ei tietenkään ole huono asia.

Taustatyön tehtyäni aloitin tietokannansuunnittelun selvittämällä, mitä tietoja tietokannan tulisi sisältää jotta toivottu toiminnallisuus saavutetaan. Tai ainakin yritin, sillä aikani tietokantatauluja pyöriteltyäni en voinut välttyä toteamukselta, että tämä reitti

osoittautui umpikujaksi. Ainakaan omalle ajattelutavalleni ei ollut mahdollista suunnitella asioita tällä tavoin. Järjellisemmältä tuntui suunnitella tietokanta siltä pohjalta, että ensin selvitin mitä käyttöliittymän tulisi sisältää, ja vasta sitten mietin mitä tietoja tietokantaan tulisi tallentaa jotta käyttöliittymä voisi näyttää tarvittavat asiat. Näin ollen päädyin suunnittelemaan käyttöliittymää ja tietokantaa rinnan.

Suunnittelutyö alkoi jälleen edetä, mutta hieman myöhemmin päädyin siihen tulokseen, että itsenäisen järjestelmän tekeminen ei ollut perusteltua. Koska alusta alkaen oli selvää, että ainakin osa järjestelmällä toteutettavista sivustoista tulisi sisältämään phpBB-foorumin, ja koska phpBB jo sinällään sisältää osan rajauksessa määritellyistä toiminnoista, oli vain loogista tehdä järjestelmästä lisäosa phpBB:lle, jo siitäkin syystä ettei yksittäisen sivuston ylläpitämiseen tarvittaisi useita käyttäjätunnuksia ja salasanoja. Lisäksi järjestelmän saattamista toimintakuntoon helpottaa huomattavasti se, että valtaosa tarvittavasta koodista löytyy valmiina tai vähintäänkin melkein valmiina, vain pieniä muutoksia vaati. Vaikka suunnitteluvaiheessa ei kuuluisikaan miettiä käytännön toteutusta (Hernandez 2000, 170), oli tämä tosiseikka liian ilmeinen jätettäväksi huomiotta, varsinkin kun se vaikuttaa itse suunnitteluun niin suuresti: onhan syntyvän järjestelmän oltava yhteneväinen niin tietokantarakenteeltaan kuin käyttöliittymältäänkin sen keskustelufoorumin kanssa, jolle se tehdään lisäosaksi, mikä ei tietenkään ole vaatimus jos siitä tehdään itsenäinen järjestelmä.

Käyttöliittymäsuunnittelun lähtökohtana oli joustavuus: ei pelkästään järjestelmän joustavuus siinä, että sen käyttäjää ei tiukasti kahlita käyttämään mitään tiettyjä, rajoitettuja rakennepohjia, vaan myös käyttöliittymän itsensä joustavuus siinä, että yksi asia voidaan tehdä useammalla kuin yhdellä tavalla. Tämä ilmenee esimerkiksi niin, että rakennepohjiin sijoitettavia tietokenttiä voi luoda omassa listassaan, ja näitä valmiiksi luotuja kenttiä sitten lisätä rakennepohjiin, mutta että kenttiä voi lisätä myös suoraan rakennepohjiin, mikä lisää kentän mahdolliseksi käyttää myös muissa rakennepohjissa. Tai niin, että yhtä tietokenttää voi käyttää useissa rakennepohjissa joko samassa tai eri tarkoituksessa, sillä nimi, jolla kenttä käyttäjälle näytetään, voidaan määrittää erikseen jokaiselle kentän ilmentymälle.

Joustavaa käyttöliittymää, jossa asiat voidaan tehdä monella tavalla, päädyin tavoittelemaan siksi, koska suurin kompastuskivi mallina käyttämissäni järjestelmissä oli mie-

lestäni juuri käyttöliittymän liiallinen jäykkyys siinä, miten asiat tuli tehdä (sen lisäksi, että ne käyttivät tarkasti määritettyjä rakenteita sisältöjen esittämiseen, mitä en halunnut). Omasta mielestäni olisi ollut huomattavasti loogisempaa tai ainakin helpompaa tehdä joitakin asioita täysin eri tavalla kuin miten ne järjestelmissä tehtiin, ja tämän asian ratkaisemiseksi on yksi vaihtoehto tarjota useampia mahdollisia tapoja tehdä jokin asia. Toinen vaihtoehto olisi tietenkin se, että toiminnot tehtäisiin toimimaan pelkästään sillä toisella tavalla, joka vaikuttaa loogisemmalta, mutta silloin ne taas voisivat tuntua vaikeammalta jonkun muun mielestä.

Koska järjestelmästä oli tarkoitus tulla lisäosa phpBB:lle, käyttöliittymä luonnollisesti tehdään sen mukaisesti. Sen on siis oltava ulkoasultaan yhtenevä muun phpBB:n kanssa, ainakin ylläpidon hallintapaneelin osalta, missä sivuston rakenteen ja muiden vastaavien asioiden määrittäminen tapahtuu. Yhteneväisyys sivuston julkisesti, tai ainakin peruskäyttäjälle ellei kaikille, nähtävässä osassa on tietysti sekin vain eduksi, jotta järjestelmä näyttää aidosti phpBB:n osalta eikä vain joltakin sen päälle liimatulta. Näin ollen käyttöliittymässä ei ole tarpeen suunnitella värejä, kirjasintyyplejä ynnä muita ulkoasun vastaavia seikkoja, jotka tulevat phpBB:n omasta ulkoasusta. Sen sijaan käyttöliittymän suunnittelussa keskitytään siihen, mitä elementtejä tulee näkyviin milläkin sivulla, miten niitä käsitellään ja miten niitä pääsee käsittelemään.

#### 4.1 Tehtävien määrittely

Tehtävien määrittelemisen edellyttää tarpeiden kartoittamista (Pohjonen 2002, 31), mutta tässä projektissa tarpeet on kartoitettu jo etukäteen, joten ne löytyvät rajauksesta. Tässä vaiheessa ei siis ole syytä alkaa enää kartoittaa tarpeita, vaan suunnittelu voidaan aloittaa heti järjestelmään tarvittavien tehtävien ja toimintojen määrittämisellä, käyttäen pohjana rajauksessa määritettyjä tarpeita.

Jotta voidaan selvittää mikä on tarpeellista, täytyy ensin tutustua ympäristöön, johon järjestelmää ollaan suunnittelemassa (Ruohonen & Salmela 2005, 65-66). Normaali tapa tähän olisi haastatella tilaavan organisaation henkilökuntaa, mutta koska tätä projektia ei tilannut mikään organisaatio, ei ollut mahdollista haastatella tilaajaorganisaation toimintaa tuntevia henkilöitä, eihän ollut ketään jota haastatella. Olisin voinut haastatella itse-



äni, mutta päätin ettei muodollisten haastattelukysymysten läpikäyntiin ole tarvetta, sillä tiesin ilman niitäkin mitä järjestelmältä haluan. Itse asiassa kysymysten listaaminen ja niihin vastaaminen olisi mahdollisesti vain tehnyt, oman ajattelutapani vuoksi, suunnittelusta vaikeampaa.

Tilaavan organisaation sijaan projektilla on kohderyhmä, joten yksi vaihtoehto, jota vakavasti harkitsin, olisi ollut haastatella kohderyhmään kuuluvia ihmisiä. Luovuin kuitenkin tästä ajatuksesta, koska saan yhteyden näihin ihmisiin vain Internetin välityksellä. Haastattelutilanteiden eteneminen olisi ollut hyvin hidasta, sikäli kuin se olisi edennyt lainkaan, sillä tällainen asia ei välttämättä mahdu kohderyhmään kuuluvien ihmisten aikatauluun, ja vaikka olisinkin lähettänyt kysymykset, en välttämättä olisi saanut niihin vastauksia projektin aikataulun puitteissa, jos olisin ylipäänsä saanut niitä koskaan. Koska kuitenkin kuulun itse kohderyhmään, ja tiesin mitä itse järjestelmältä haluan, päädyin ratkaisuun, jossa suunnittelin toiminnot ensisijaisesti omien tarpeideni pohjalta, ja yleistin mahdollisuuksien mukaan ratkaisut ja termit myös muille kohderyhmän jäsenille, ja mahdollisesti myös sen ulkopuolisille ihmisille, sopiviksi.

Aloitin ympäristöön tutustumisen tutkimalla ja analysoimalla kohderyhmään kuuluvien ihmisten, myöskin itseni, ylläpitämien sivustojen sisältöä ja rakenteita. Sivujen valikointiin käytin omaa henkilökohtaista linkkilistaani, joka sisältää muutaman kymmentä sivustoa, mutta koska en ole saanut sivustojen omistajilta lupaa omien sivujensa mainintaan tässä yhteydessä, jätän ne luetteloimatta. Useimmat sivustoista ovat niin samankaltaisia, että yleistyksiä tarpeellisista rakenteista on helppo tehdä, mutta jotkin yksilöllisemmät, monimutkaisemmalla sivustorakenteella toteutetut sivustot eivät sopineet näihin yleistyksiin (taulukko 1). Näiden tulosten perusteella päätin, minkälaisia rakenteita järjestelmän pitää pystyä hallinnoimaan.

Tutkittuani sisällönhallintajärjestelmiä ja kohderyhmän sivustoja selvitin, mitä ominaisuuksia suunniteltavan sisällönhallintajärjestelmän tulee tarjota sivuston hallinnalle. Tutkimusaineiston lisäksi ominaisuuksien määrittämisessä tuli ottaa huomioon myös projektin rajaus, joka jättää osan web-sisällönhallintajärjestelmien yleisistä ominaisuuksista suunniteltavan järjestelmän tehtävien ulkopuolelle (taulukko 2).

Taulukko 1. Vertailu kohderyhmään kuuluvien eri sivustojen tyypillisimmistä osioista. Kuten taulukosta käy ilmi, monet osiot ovat samanlaisia eri sivustoissa, vaikka ne poikkeaisivatkin toisistaan nimeltään.

<b>Sivusto 1</b>	<b>Sivusto 2</b>	<b>Sivusto 3</b>
Etusivu	Alku	Etusivu
Sivuston etusivu, sisältää tervetuloitovotuksen ja uusimmat päivitykset.	Sivuston etusivu, sisältää tervetuloitovotuksen ja uusimman päivityksen ajankohdan.	Sivuston etusivu, sisältää lyhyen infotekstin, valittuja linkkejä ja uusimman päivityksen ajankohdan.
Sarjakuva	Tarinat	Projektit
Sivuston pääaiheena toimivan sarjakuvan lukusivu.	Sivuston pääaiheena toimivien tekstimuotoisten tarinoiden arkistosivu.	Sivuston pääaiheena toimivat tarinat, joista osa on sarjakuva- ja osa tekstimuotoisia.
Hahmot	Henkilöt	
Tietoa sarjakuvan henkilöistä.	Tietoa tarinoiden henkilöistä.	Projekteilla on alasivuja, jotka sisältävät tietoa tarinoiden henkilöistä, tarinoiden lukusivut, lisätietoa tarinoiden maailmoista ynnä muuta.
Info	Lisätietoa	
Lisätietoa sarjakuvan maailmasta.	Lisätietoa tarinan maailmasta.	
Galleria	Kuvasto	Galleria
Sivu, jolla on ylläpitäjän tekemiä kuvia; ryhmitelty valmistumisvuoden mukaan.	Sivu, jolla on ylläpitäjän tekemiä kuvia; kuvat ovat aikajärjestyksessä, mutta niitä ei ole luokiteltu aliryhmiin.	Sivu, jolla on ylläpitäjän tekemiä kuvia; ryhmitelty aihepiireittäin.
Linkit	Linkit	Linkit
Linkkejä muille sivustoille, ei ryhmittelyä.	Linkkejä muille sivustoille, ryhmitelty aihepiireittäin.	Linkkejä muille sivustoille, ryhmitelty aihepiireittäin.

Vieraskirja	Vieraskirja	Vieraskirja
Sivuston vieraskirja, yhteydenottoväline ylläpitäjään.	Sivuston vieraskirja, yhteydenottoväline ylläpitäjään.	Sivuston vieraskirja, yhteydenottoväline ylläpitäjään.
Ylläpitäjä	Webmaster	Ylläpito
Tietoa sivuston ylläpitäjästä.	Tietoa sivuston ylläpitäjästä.	Tietoa sivuston ylläpitäjästä.
Fanitaide	Vierastaide	Kävijöiltä
Kävijöiden tuotoksia sivuston tarinoista.	Kävijöiden tuotoksia sivuston tarinoista.	Kävijöiden tuotoksia sivuston tarinoista.

Taulukko 2. Vertailu Joomla!n (lähde: Joomla.org, Features overview) ja phpBB:n (lähde: phpBB Group, Features of phpBB) ominaisuuksista sekä ACMS:n tarvitsemista toiminnoista.

<b>Joomla!</b>	<b>phpBB</b>	<b>ACMS</b>
Käyttäjien hallinta	Käyttäjien hallinta	
	Yksityisviestit	
Median hallinta	Liitetiedostojen hallinta	Kuvien hallinta
Kielten hallinta	Kielten hallinta	
Bannerien hallinta	(mainosten hallintaan on tarjolla lisäosa)	
	Valvontaominaisuudet	
Sisällön hallinta	Viestien hallinta	Tekstisisällön hallinta
		Etusivun hallinta
		Uutisten hallinta
Kyselyt	Kyselyt	
Hakutoiminto	Hakutoiminto	

Linkkien hallinta	(linkkien hallintaan on tarjolla lisäosa)	Linkkien hallinta
Syötteiden hallinta	Syötteiden hallinta	
Valikkojen hallinta	(rajoitettu mahdollisuus hallita valikkoja)	Valikkojen hallinta
	Keskustelualueiden hallinta	Tietokokonaisuuksien hallinta
Tyylipohjien hallinta	Tyylien hallinta	Tyylien hallinta
Integroitu ohjetoiminto	Ohjetoiminto	
Sivuston asetukset	Foorumin asetukset	Sivuston asetukset
Web-palvelut	Lokitietojen keruu	Lokitietojen keruu
Lisäosat	Lisäosat	Omat tietokentät

Perinteisten dokumenttienhallintajärjestelmien tärkeimmät ominaisuudet ovat dokumenttien luonti, ylläpito, haku, katselu ja tulostus. Tietokantaan tallennetaan dokumenttien ominaisuustietoja, joiden perusteella dokumentteja myös haetaan. Dokumentteja luokitellaan ominaisuustietojen ja kansiorakenteiden perusteella ja niiden eri versioita hallinnoidaan. Dokumenttien hallintaan kuuluu myös käyttöoikeuksien hallinta ja dokumenttien päällekkäisten muokkausten estäminen. (Anttila 2001, 20)

Suunnitteilla olevan web-sisällönhallintajärjestelmän rajauksen mukaan sisältöä voi luoda vain yksi käyttäjä, ja kaikki sisältö on kaikkien nähtävissä, joten käyttöoikeuksien hallintaa ja dokumenttien päällekkäisten muokkausten estämistä ei tarvita. Rajaus ei myöskään sisällä hakutoimintoa, ja koska sisältö on tarkoitus esittää vain web-selaimessa, tulostustoimintoakaan ei tarvita, vaikka se toisaalta voisikin olla hyödyllinen ja näin ollen tarpeen lisätä myöhemmin. Erilaisia tietokokonaisuuksia on kuitenkin voitava luoda, ylläpitää ja katsella. Tiedot talletetaan tietokantaan, ja varsinaisten sisältöjen lisäksi tallennetaan niitä kuvaavia ominaisuuksia.

Joomla! tarjoaa ylläpitäjälle toiminnot artikkelien ja staattisen sisällön hallintaan, aiheiden ja kategorioiden hallintaan, valikoiden ja käyttäjien hallintaan, kuvien ja videoiden

hallintaan, etusivun hallintaan ja sivuston asetusten hallintaan. Lisäksi Joomla! mahdollistaa tyylien hallinnan ja erilaiset liitännäiset, kuten mainosten hallinnoinnin ja tuoreimpien uutisten näytön etusivulla. (Joomla.org, Features overview)

Projektissa suunniteltavan järjestelmän ei ole tarkoitus olla heti yhtä monipuolinen kuin Joomla!, jota on kehitetty jo pitkään, sillä se veisi paljon enemmän suunnittelu-aikaa. On kuitenkin joitakin osia, joita suunnittelussa voidaan hyödyntää. Suunniteltavan järjestelmän on myös tarpeen voida hallita tekstisisältöä, joskaan jakoa artikkeleihin ja staattiseen sisältöön ei välttämättä ole tarpeen tehdä. Kategorioita on hallittava, sillä kohderyhmän sivustot jakautuvat usein eri osa-alueisiin. Lisäksi on tarpeen hallita valikkoja, kuvia ja sivuston asetuksia, tyylejä ja etusivua. Mainoksia ei kohderyhmän sivustoilla yleensä käytetä, joten niiden hallinta ei ole tässä vaiheessa tarpeen, vaikka sellainen toiminto voitaisiinkin toteuttaa myöhemmin. Käyttäjää ei myöskään ole tarpeen hallita, kuten edellä jo todettiin. Uutisten hallinnoinnin tarpeellisuutta sen sijaan voidaan harvita, sillä sivustoilla useimmiten ilmoitetaan tuoreimmat päivitykset.

Keskustelufoorumisovellus phpBB tarjoaa työkalut käyttäjien ja alueiden hallintaan sekä tyylien ja erilaisten palvelimen ja keskustelufoorumien asetusten, esimerkiksi sähköpostitoimintojen ja rekisteröitymisen, hallintaan. Lisäksi tarjolla on runsaasti erilaisia lokitietoja foorumin tapahtumista. Järjestelmä tarjoaa kuitenkin työkaluja myös omien tietokenttien ja liitetiedostojen, esimerkiksi kuvien, hallintaan. Mahdollista on myös muokata tyylejä sovelluksen kautta, ilman että tarvitsee avata tiedostoja erikseen toiseen ohjelmaan. Erityisesti ulkoasussa käytettyjen kuvien vaihtaminen on helppoa, sillä ne saadaan ladattua palvelimelle ja vaihdettua parilla napin painalluksella. (phpBB Group, Features of phpBB)

Omien tietokenttien hallinta on tarpeellinen ominaisuus suunnitteilla olevassa järjestelmässäkin, mutta sen lisäksi aiemmin listattujen toimintojen joukkoon phpBB ei tuo varsinaisesti uusia. Kyseisen järjestelmän toiminnot ovat kuitenkin osin helppokäyttöisempiä kuin Joomla!-järjestelmässä, joten niiden toiminnallisuutta on parempi käyttää esikuvana uuden järjestelmän toiminnallisuutta suunniteltaessa. Erityisesti ulkoasun kuvien hallinnan helppous on esimerkillistä ja kannattaa ottaa huomioon (kuva 2). Suunniteltavassa järjestelmässä on lisäksi hyödyllistä kerätä myös jonkinlaisia lokitietoja, ainakin sivulatausten määriä.

Tarvittavat toiminnot ovat siis seuraavat:

- tietokokonaisuuksien hallinta: kategorioiden ja sivujen luonti, muokkaus ja poistaminen
- tekstisisällön hallinta: tekstien luonti, muokkaus, poistaminen ja liittäminen kategorioihin ja sivuihin
- uutisten hallinta: uutisten luonti, muokkaus ja poistaminen, uutisisältöjen liittäminen sivuihin
- valikkojen hallinta: valikkoelementtien määrittäminen, muokkaus ja poistaminen
- kuvien hallinta: kuvien luonti, muokkaus ja poistaminen, kuvien liittäminen kuvagallerioihin
- sivuston asetusten hallinta
- tyylien hallinta: tyylien lisäys, muokkaus ja poistaminen
- etusivun hallinta, etusivun elementtien muokkaaminen
- omien tietokenttien hallinta: tietokenttien luonti, muokkaus ja poistaminen, tietokenttien liittäminen tietokokonaisuuksiin ja sivuihin
- lokitietojen keruu ja esittäminen

Toimintojen tarkemmat kuvaukset on kirjattu liitteeseen 2.

Kuva 2. phpBB:n kuvien hallinta. Ylemmästä pudotusvalikosta (”Select image”) valitaan, mitä tyyli pohjan kuvaa vaihdetaan; alemmasta pudotusvalikosta (”Image”) valitaan, mitä foorumin kuvakansioon tallennetuista kuvatiedostoista halutaan käyttää. Va-

linnan tekeminen pudotusvalikosta päivittää esikatselukuvan ("Selected image"). Valinta otetaan käyttöön lähettämällä lomake ("Submit"). phpBB on saatavilla myös suomeksi, mutta kuvan tilanteessa foorumilla oli käytössä phpBB:n oletuskieli englanti.

## 4.2 Ensimmäinen järjestelmä

Suunnittelun aloitusvaiheessa työstä oli tarkoitus tehdä erillinen, itsenäinen järjestelmä. Ensimmäisen järjestelmän suunnitelma sisältää tietokannan suunnittelun, osittaisen käyttöliittymän suunnittelun ja käyttöliittymätestauksen suunnittelun. Koska kesken kaiken tämän järjestelmän suunnittelua kuitenkin päädyin siihen, että itsenäisen järjestelmän tekeminen ei ole järkevää, tätä versiota ei koskaan suunniteltu valmiiksi asti. Se on kuitenkin olennainen osa työn suunnittelua ja osa toisen järjestelmän rakenteesta on johdettu suoraan siitä, joten se esitellään tässä.

### 4.2.1 Tietokannan suunnittelu

Tiedon tallennuspaikkana tietokanta on koko järjestelmän pohja, joten järjestelmälle on suunniteltava sellainen tietokanta, joka mahdollistaa järjestelmän toiminnan ja tukee sen tarkoitusta. Siihen on voitava tallentaa määriteltyjen ominaisuuksien mukaiset tiedot, joiden lisäksi siihen on tallennettava myös toiminnallisuuden aikaansaamiseksi tarvittavat tiedot. Jälkimmäiset eivät välttämättä sivuston selaajalle näy missään, ja sivuston ylläpitäjällekin vain mikäli tämä tutkii tietokantaa suoraan.

Aloitin tietokannan suunnittelun selvittämällä, mitä tietoja tietokantaan on tallennettava, jotta tietokannan sisällön perusteella voidaan esittää sivusto. Aluksi keskityin tutkimaan sitä, minkälaisia tietoja kohderyhmän sivustoilla eri elementteihin, esimerkiksi linkkeihin, kuuluu. Nämä sivustot koostuvat pääasiassa erilaisista kuvagallerioista, hahmoesitelyistä, sarjakuvaprojekteista ja linkkilistoista, joten tietokantaan tallennettavia tietoja ovat pääasiassa nimet, osoitteet, pidemmät ja lyhyemmät kuvaukset ja muut eri tavoin ryhmitellyt tekstit (taulukko 3).

Taulukko 3. Luettelo tyypillisimmistä nuorten harrastelijaylläpitäjien sivustojen elementeistä jaoteltuna osiin, joista ne koostuvat.

<b>Elementti</b>	<b>Sisältö</b>	<b>Tyyppi</b>
Sivu	Otsikko	tekstirivi
	Alaotsikko	tekstirivi
	Teksti	tekstikappale
Päivitys	Ajankohta	päivämäärä
	Päivitysteksti	tekstikappale
	Luettelo tapahtumista	teksti/luettelo
Linkki	Osoite	URL
	Nimi	tekstirivi
	Vaihtoehtoinen teksti	tekstirivi (ei erikoismerkkejä)
	Banneri	kuvan URL
	Kohdeikkuna	määre (sana)
Linkkiryhmä	Nimi	tekstirivi
	Linkistö	kooste linkeistä
Kuvagalleria	Nimi	tekstirivi
	Kuvasto	kooste kuvalinkeistä
Kuva	Nimi	tekstirivi
	Kuvaus	tekstikappale
	Tekoaika	päivämäärä
	Työvälineet	teksti/luettelo
	Oikeudet	teksti/luettelo
Sarjakuva	Otsikko	tekstirivi
	Arkisto	kooste sarjakuvasivuista
Sarjakuvasivu	Kuva	kuvan URL



	Sivunumero	luku/tekstirivi
	Siirtymislinkit	linkki
Teksti	Otsikko	tekstirivi
	Kuvaus	tekstikappale
	Teksti	monta tekstikappaletta

Suunnittelussa keskityin aluksi siihen, että tieto tallennettaisiin tällaisten sivustoille ominaisten rakennepohjien mukaan, niin että jokainen eri rakennepohja tallennettaisiin omana tietokantataulunaan. Etuna tässä mallissa on, että sivustoa esitettäessä tietokantatauluista on helppoa etsiä oikeanmuotoista tietoa. Haittana on, että tällöin pitäisi luoda uusi tietokantataulu aina, kun luodaan uusi rakennepohja. Tästä mallista on kaava liitteessä 3.

Koska järjestelmän on valmiiden rakennepohjien hyödyntämisen lisäksi tarkoitus mahdollistaa myös niiden muokkaaminen, lisääminen ja poistaminen, tulin siihen tulokseen, että oman taulun luominen jokaiselle rakenteelle ei ollut tyydyttävä ratkaisu. Tauluihin olisi luotava ylimääräisiä kenttiä, jotta voitaisiin yrittää varautua ylläpitäjän rakenteisiin haluamiin muutoksiin, mutta on mahdotonta ennakoida kaikkia niitä muutoksia joita tuntemattomat ylläpitäjät voisivat tulevaisuudessa tahtoa pohjiin tehdä. Voitaisiin tietenkin asettaa rajat pohjan muokkaukselle, mutta järjestelmän on tarkoitus olla mahdollisimman joustava, eikä tällaisten rajoitusten asettaminen sovi joustavaan järjestelmään. Toinen vaihtoehto olisi luoda uutta rakennepohjaa luotaessa aina uusi taulu, ja rakennepohjaa muokattaessa muokata tietokantataulun rakennetta. Tuloksena olisi monimutkainen toteutus, jossa olisi lukuisia lähes samanlaisia tietokantatauluja ja monia keskenään samaa asiaa kuvaavia tietokenttiä.

Koska en päässyt eteenpäin valmiiden rakennepohjien taulurakenteiden selvittämisessä, aloin pohtia ylläpitäjän omien rakennepohjien hallinnointiin tarvittavia taulurakenteita. Tässä käytin pohjana phpBB3:n tietokantarakennetta, mainitussa järjestelmässä kun on käytössä toiminto, jossa ylläpitäjä voi luoda käyttäjäprofiileihin haluamansa määrän haluamansa tyyppisiä tietokenttiä valmiiksi määriteltyjen tietokenttien lisäksi (kuva 3). Arvelin, että sisällönhallintajärjestelmän mukautetut rakenteet voisi tallentaa siten, että

yhteen tauluun tallennetaan kenttien tyypit ja toiseen niiden sisällöt. Nämäkin tosin käytännössä toteutetaan phpBB:ssa siten, että mukautetut kentät luodaan erikseen kenttinä niille varattuun tietokantatauluun, mutta omassa järjestelmässäni niitä ei tarvitse silti luoda tällä tavoin, vaikka vaikutteita olisikin otettu phpBB:n järjestelmästä.

Aikani pohdiskeltuani mukautettujen rakenteiden tallennustapaa tulin siihen tulokseen, että myös valmiiksi luodut rakennepohjat voisi tallentaa samalla tavalla. Tällöin voidaan muokata helposti paitsi ylläpitäjän omia myös valmiita rakennepohjia, ja valmiiden pohjien mukaan voidaan helposti luoda myös uusia rakennepohjia muokkaamalla vain tietokantataulujen sisältöä, ei niiden rakennetta. Tässä ratkaisussa on edelleen omat taulunsa erityyppisille sisällöille, mutta sisällöt on jaettu pienempiin, toisistaan erottamattomiin osiin, kuten esimerkiksi web-linkissä kohteen nimi ja sen URL-osoite. Rakennepohjat itsessään tallennetaan toiseen tauluun, johon kirjataan rakenteessa esiintyvien kenttien tyypit, nimet ja järjestys. Tämä taulurakenne mahdollistaa suuremman joustavuuden, joskin sivujen muodostaminen muuttuu hieman vaikeammaksi, sillä ennen kuin voidaan etsiä itse esitettävä tieto, on etsittävä tieto esitettävän sivun rakenteesta. Turhilta tauluilta ja lukuisilta samaa asiaa kuvaavilta kentiltä kuitenkin säästytään.

Kummallakin vaihtoehdolla sivuston sisällön tallentamiseen on omat heikkoutensa ja vahvuutensa, mutta päädyin kuitenkin lopulta jälkimmäiseen vaihtoehtoon pääasiassa sen joustavuuden vuoksi. Koska rakenteiden muokkaus on olennainen osa järjestelmää, päättelin, että joustava tietokantarakenne on parempi vaihtoehto kuin joustamaton. Jälkimmäisessä vaihtoehdossa on lisäksi etuna se, että muokatut rakenteet ja valmiit rakenteet eivät tallennustavaltaan poikkea toisistaan, joten ne voidaan myös esittää samalla tavalla. Kaikki rakenteet ovat tasa-arvoisia, eikä ylläpitäjän tarvitse välittää valmiista rakennepohjista ellei hän niin halua.

Pääavaimeksi on tauluihin luotava juoksevat numerot, sillä tauluissa ei itsessään ole mitään sellaista saraketta, joka olisi pääavaimeksi sovelias. Nimet, otsikot ja osoitteet voivat toistua useaan kertaan saman sivuston tiedoissa, ja kuvaustekstit ovat liian pitkiä toimiakseen avaimina. Viiteavaimia tarvitaan erityisesti rakenteita ja sisältöjä kuvaavien taulujen välille, mutta myös osoittamaan rakenteiden suhteita toisiinsa.

## Edit profile field

« Back

Here you can change the basic parameters of your profile field. The relevant options are re-calculated within the second step.

**Edit profile field**

<p><b>Field type:</b> You are not able to change the field type later.</p> <p><b>Field identification:</b> The field identification is a name to identify the profile field within the database and the templates.</p> <p><b>Publicly display profile field:</b> The profile field will be shown in all locations allowed within the load settings. Setting this to "no" will hide the field from topic pages, profiles and the memberlist.</p>	<p><b>Boolean (Yes/No)</b></p> <p><b>identor</b></p> <p><input checked="" type="radio"/> Yes <input type="radio"/> No</p>
---	---

**Visibility options**

<p><b>Display in user control panel:</b> The user is able to change this profile field within the user control panel.</p> <p><b>Display on registration screen:</b> If this option is enabled, the field will be displayed on registration.</p> <p><b>Display on viewtopic screen:</b> If this option is enabled, the field will be displayed in the mini-profile on the topic screen.</p> <p><b>Required field:</b> Force profile field to be filled out or specified by user or administrator. If display at registration screen option is disabled, the field will only be required when the user edits their profile.</p> <p><b>Show field if no value was selected:</b> Determines if the profile field should be displayed if no value was selected for optional fields or if no value has been selected yet for required fields.</p> <p><b>Hide profile field:</b> Hide the profile field from all other users except the user, administrators and moderators who are still able to see this field. If the Display in user control panel option is disabled, the user will not be able to see or change this field and the field can only be changed by administrators.</p>	<p><input checked="" type="checkbox"/></p> <p><input checked="" type="checkbox"/></p> <p><input checked="" type="checkbox"/></p> <p><input type="checkbox"/></p> <p><input type="checkbox"/></p> <p><input type="checkbox"/></p>
--	--

**Language specific options [fi]**

<p><b>Field name/title presented to the user:</b></p> <p><b>Field description:</b> The explanation for this field presented to the user.</p> <p><b>Entries:</b> Enter your options now</p>	<p>Identor</p> <p>Oletko koira vai kissa?</p> <p>koira First option</p> <p>kissa Second option</p>
--	--

## Profile type specific options

« Back

Here you are able to change some common options.  
**Please note that changes to profile fields will not affect existing profile fields entered by your users.**

**Profile type specific options**

<p><b>Field type:</b> Define the type, either a checkbox or radio buttons. A checkbox will only be displayed if it is checked for a given user. In that case the <b>second</b> language option will be used. Radio buttons will display regardless of their value.</p> <p><b>Default value:</b></p>	<p><input checked="" type="radio"/> Radio buttons <input type="radio"/> Checkbox</p> <p><input type="radio"/> koira <input checked="" type="radio"/> kissa</p>
---	--

Kuva 3. phpBB:n tapa luoda ja muokata käyttäjäprofiilin mukautettuja kenttiä. Kentän tyyppi valitaan pudotusvalikosta kun se luodaan, luonnin jälkeen tyyppiä ei ole enää mahdollista muuttaa. Yleiset asetukset määritetään omalla sivullaan, napin ”Profile type specific options” painaminen tuo esiin sivun, jossa nimensä mukaisesti hallitaan kentän tyyppin mukaisia asetuksia. Tässä tapauksessa kyseessä on kyllä/ei -arvo.

Sivuston rakennetta ja sisältöä kuvaavien taulujen lisäksi tarvitaan tietokantaan vielä muitakin tauluja. Näillä tauluilla kuvataan yleisiä ja erityisiä asetuksia jotka koskevat koko sivustoa, eivät vain sen yksittäisiä sivuja. Kun olin saanut selvitettyä sivuston sisältöjen tallentamiseen liittyvät ongelmat, siirryin selvittämään sivuston yleisten asetusten ja rakenteisiin liittyvien aputaulukojen edellyttämiä tietoja ja rakenteita. Nämä tiedot käsittävät sellaisia asioita kuin sivuston nimi, käytössä oleva ulkoasu ja osoite, ynnä muita sivuston kokonaisuuden kannalta oleellisia asetuksia.

Sivuston asetusten tallentamisen taulurakenteiden suunnittelun aloitin myös phpBB:n asetusten taulurakenteesta, jossa asetukset on tallennettu tauluun siten, että sarakkeita on vain kaksi: ”nimi” ja ”arvo”. Rivejä on niin monta kuin on tallennettavia asetuksiakin. Koska järjestelmällä on tässä vaiheessa tarkoitus hallinnoida vain yhtä sivustoa, tällainen menetelmä saattaisi toimia. Jos jokaiselle arvolle tehdään tauluun oma sarakkeensa, yhtä sivustoa hallinnoivan järjestelmän tietokantatauluun tulee vain yksi tietue. Tässä mallissa pääavaimena toimii asetuksen nimi, sillä yhdessä sivustossa kutakin asetusta on olemassa vain yksi.

Järjestelmää on kuitenkin tarkoitus tulevaisuudessa laajentaa siten, että sillä voidaan hallinnoida samanaikaisesti useita sivustoja, ja tällöin on parempi että eri arvoille on eri sarakkeet. Silloin tauluun voidaan tallentaa yhden sivuston asetukset yhteen tietueeseen, mikä on selkein ratkaisu jos sivustoja on useita. Tulin siihen tulokseen, että on järjetöntä suunnitella järjestelmään tällainen osa tavalla, joka tullaan joka tapauksessa muuttamaan täysin, jos se voidaan suunnitella myös tavalla, joka ei tarvitse kovin suurta muokkaamista järjestelmää laajennettaessa. Näin ollen päädyin tekemään jokaiselle asetukselle oman sarakkeensa siitäkin huolimatta, että tauluun tulee tässä vaiheessa vain yksi tietue. Tällöin voidaan myös hyödyntää viiteavaimia, mitä ei voitaisi tehdä silloin, kun erityyppiset asetukset ovat kaikki samassa kentässä. Tässä mallissa pääavaimena voisi toimia joko sivuston nimi tai sitten nimenomaan pääavaimeksi luotu juokseva numero. Päädyin tässä vaiheessa käyttämään tarkoitukseen sivuston nimeä.

Tarkempi selvitys tietokantarakenteen vaihtoehtoista löytyy liitteistä 4 ja 5.

#### 4.2.2 Käyttöliittymän suunnittelu

Käyttöliittymän suunnittelun tavoitteena on saada aikaan käyttöliittymä, joka mahdollistaa tietojen tallentamisen tietokantaan ja esittää tietokantaan tallennetut tiedot niin, että sivuston ylläpitäjä kykenee hallinnoimaan sivustoa helposti ja vaivattomasti. Käyttöliittymän on mahdollistettava sekä tietojen esittäminen että niiden lisääminen, poistaminen ja päivittäminen. Lisäksi käyttöliittymän tulee olla yksinkertainen käyttää ja miellyttävä tai vähintäänkin voimakkaita negatiivisia tunteita herättämätön katsella, jotta sen käyttäminen on mahdollista ilman että käyttäjän täytyy ponnistella vastustamaan halua paiskata koko järjestelmällä vesilintua.

Aloitin käyttöliittymän suunnittelun tutkimalla verkkosisällön hallintaan tarkoitettujen Open Source -periaatteella kehitettyjen järjestelmien käyttöliittymiä. Tutkin pääasiassa kahta järjestelmää, joista toinen oli Joomla! ja toinen phpBB, kuten tehtäviä määrittäessänikin. Näissä järjestelmissä on toteutettu samantyyppisiä asioita erilaisilla ratkaisuilla, joista toiset tuntuivat intuitiivisemmilta käytettäviltä kuin toiset. Tutkittuani toteutuksia ja pohdittuani mikä valinnoissa on onnistunut, mikä vähemmän onnistunut, sovelsin käytettyjä ratkaisuja omissa suunnitelmissani ylläpitoliittymän rakenteeseen, toimintojen hallinnointiin ja tallennettujen tietojen esittämiseen. Tärkeimmät huomiot tästä vaiheesta on kirjattu taulukkoon 4.

Taulukko 4. Vertailu tärkeimpien Joomla!-n ja phpBB:n toimintojen toteutustavoista.

Ominaisuus	Joomla!	phpBB
Ylläpitoliittymän etusivu	Eri toimintojen kuvakkeita + muutama luettelo tapahtumista ja artikkeleista.	Tilastoja foorumista + muutama luettelo tapahtumista ja käyttäjistä.
Valikkorakenne	Monitasoiset pudotusvalikot.	Välilehdet, alemman tason valikot sivupaneelissa (jonka voi sulkea).
Sisällön rakenne	Valikot + artikkelikategoriat + moduulit.	Keskustelualueet + valikot.

Rakenteen hallinta	Valikot, kategoriat ja moduulit on kaikki asetettava erikseen ja erillisellä toimenpiteellä yhdistettävä.	Keskustelualueet kuuluvat automaattisesti toistensa sisään, mutta niille on asetettava käyttöoikeudet ennen kuin ne näkyvät. Hallintapaneelien valikot määritetään erillisessä moduuliosassa, foorumin muut valikot asetetaan foorumin asetuksissa tai tyyli-pohjissa.
Sisällön hallinta	Artikkeleja voi lisätä sekä sivustolta että ylläpidon hallintapaneelista käsin.	Viestejä voi lähettää vain foorumilla, ei ylläpidon hallintapaneelissa.
Tyylin hallinta	Valikot asetetaan tyylikoh- taisesti.	Tyylikohtaisia asetuksia ei ole (lukuun ottamatta tyylin omaa kuvavalikoimaa, joka on osa tyyliä).
Ulkoasujen muokkaus	Ulkoasu sisältää php-koodia.	Ulkoasu sisältää HTML- ja CSS-koodia, minkä lisäksi siinä on erityisiä muuttujia.
Komentopainikkeet	Komentopainikkeet ovat sisällön yläpuolella.	Komentopainikkeet ovat niiden osien vieressä tai alapuolella, joita niillä käsitellään.
Käyttäjien hallinta	Käyttäjiä voi lisätä hallintapaneelista.	Käyttäjiä voi lisätä hallintapaneelista vain lisäosan avulla.

Järjestelmään talletettujen tietojen muodostaman julkisesti esitettävän sivuston ulkoasu tulee olemaan ylläpitäjän vapaasti määritettävissä, joten en perehdy tässä yhteydessä sen käyttöliittymän suunnitteluun. Sen sijaan keskityn ylläpitoliittymän ulkoasun ja käyttöliittymän suunnitteluun. Tämäkin ulkoasu voidaan vaihtaa tulevassa järjestelmässä, jos

niin halutaan, mutta ylläpitoliittymän tulee olla toimiva ja käytettävä myös ilman, että sivuston ylläpitäjä panostaa aikaansa sen suunnittelemiseen. Sivustoa ei voi ylläpitää, ellei ylläpitoliittymää pysty käyttämään, ja todennäköisesti pidetään tärkeämpänä muokata julkisesti esitettävän sivuston kuin ylläpitoliittymän ulkoasua, varsinkin ellei ylläpitäjällä ole ruhtinaallisesti aikaa ja ellei ylläpitoliittymä ole yksinkertaisesti niin epämiellyttävä, ettei ylläpitäjä tahdo katsella sitä sekuntiakaan ylimääräistä.

Kyseessä olevan järjestelmän käyttöliittymän on oltava visuaalisesti mielenkiintoinen, sillä sen kohderyhmä arvostaa visuaalisuutta. Toisaalta käyttämisen on kuitenkin oltava selkeää ihmisille, jotka eivät ole harjaantuneet käyttämään sisällönhallintajärjestelmiä, koska todennäköisesti suurin osa kohderyhmään kuuluvista ei ole koskaan käyttänyt monimutkaista sähköistä sisällönhallintaa. Toiminnan ja rakenteen on siis oltava mahdollisimman helposti ymmärrettävää. On siis mahdollisuuksien mukaan pyrittävä yhdistämään taiteellisuuteen ja käytännöllisyyteen pyrkivät päämäärät.

Ulkoasun visuaalisuuden voi toteuttaa grafiikalla, mutta sivustosta voi saada visuaalisesti mielenkiintoisen myös pelkillä värialueilla. Päädyin käyttämään visualisointiin väritettyjä taulukoita ja lohkoelementtejä, sillä tällä menetelmällä sivuista tulee kevyempiä, koska palvelimelta ladattavien tiedostojen yhteinen tiedostokoko on pienempi ilman kuvia joten sivu latautuu nopeammin (Nielsen 2000, 46). Erityisen näyttävää sivustoa ei tosin tällä tekniikalla saada aikaan, mutta ylläpitoliittymän on kuitenkin tarkoitus olla vain sivuston hallinnoinnin apuväline. Raskaat, paljon grafiikkaa sisältävät sivut kuluttavat paljon tiedonsiirtoon varattua kaistaa, ja sen määrä on web-hotelleissa usein rajattu. On siis parempi säästää mahdollisimman paljon kaistaa julkisesti esitettävälle sivustolle, varsinkin kun nuorten harrastelijaylläpitäjien sivustot sisältävät usein runsaasti kuvallista materiaalia.

Ylläpitoliittymän värit pyrin valitsemaan mahdollisimman rauhalliseksi, jotta sivuston hallinnointi olisi silmille miellyttävää. Päädyin vaaleaan sinisävyiseen taustaväriin, sillä sininen on rauhallinen väri (Rihlma 1997, 109) ja painuu laajoina pintoina taustalle, joten se soveltuu hyvin taustaväriksi, sillä kaukainen maisemakin näyttää aina sinertävältä. Aluksi harkitsin taustaksi vaaleaa neutraalia harmaata, mutta koska itse saan jos-takin itselleni tuntemattomaksi jääneestä syystä aina päänsärkyä lukiessani muutamaa riviä pidempää tekstiä harmaalta taustalta, päätin että kyseinen väri ei ole tähän tar-

koitukseen sopiva, tuleehan ylläpitäjä tuijottaneeksi ylläpitoliittymää tovin jos toisenkin. Tekstiksi selkeä ja perinteikäs valinta olisi musta, joka on paitsi tavanomaisin kirjoitetun tekstin väri myös taustasta hyvin erottuva, mutta väriskaalan harmonisuuden vuoksi valitsin erittäin tumman sinisen. Lisäksi tarvitaan huomioväri, jolla ilmoitetaan tärkeistä asioista ja virhetilanteista, ja tähän tarkoitukseen valitsin tavanomaisen punaisen siitäkin huolimatta, että se sointuu sinisen kanssa hieman huonosti yhteen. Punainen on kuitenkin yleinen väri virhetilanteiden ilmoittamiseen, joten arvelin sen soveltuvan tähän käyttöön nytkin. Ylläpitoliittymän malli löytyy kuvasta 4.

**ACMS :: ylläpitoliittymä**

Kirjaudu ulos | Etusivu | Asetukset | Rakenteet | Kategoriat | Linkit | Kuvat | Sisällöt | Tyylit

## Sivuston kategoriat

### Pääkategoriat

**Kategorian 1 nimi** - avaa | muokkaa | poista | siirrä ylöspäin | siirrä alaspäin  
 Kategorian 1 kuvaus tulee tähän.  
 Alakategorioita: 2 | Tekstejä: 5 | Hahmoja: 2 | Kuvia: 3

**Kategorian 2 nimi** - avaa | muokkaa | poista | siirrä ylöspäin | siirrä alaspäin  
 Kategorian 2 kuvaus tulee tähän.  
 Alakategorioita: 0 | Tekstejä: 0 | Hahmoja: 2 | Kuvia: 0

**Kategorian 3 nimi** - avaa | muokkaa | poista | siirrä ylöspäin | siirrä alaspäin  
 Kategorian 3 kuvaus tulee tähän.  
 Alakategorioita: 0 | Tekstejä: 0 | Hahmoja: 2 | Kuvia: 0

Lisää uusi:

ACMS is copyright © by Liisa Peippo 2009  
 this site is for testing use only

Kuva 4. ACMS-ylläpitoliittymän malli. Kuvassa ollaan ”Sivuston kategoriat”-sivulla, jolla hallitaan sivuston kategorioita eli valikon päätason otsikkoja. Valikkopalkin taustaväri toimii myös virheilmoituksen taustavärinä virheen sattuessa.

Sisällönhallintajärjestelmä on tyypiltään ekstranet-järjestelmää muistuttava, sillä käyttäjä on jo sitoutunut käyttämään järjestelmää asentaessaan sen. Käyttäjää ei siis tarvitse enää houkutella järjestelmän käyttäjäksi, kuten web-sivuilla vierailevia on houkuteltava jotta nämä kävisivät sivustolla toistekin. Sisällönhallintajärjestelmääkin voidaan kyllä vaihtaa, ellei siihen olla tyytyväisiä, mutta todennäköisesti tällaista vaihtamista ei tehdä kovin kevyin perustein, ainakaan sen jälkeen kun järjestelmän kokeiluvaiheesta on päästy ohi ja sen varsinainen käyttö aloitettu. Siitä huolimatta on tarpeen kiinnittää huomiota käytettävyyteen, jotta käytön opettelu ei osoittaudu liian haasteelliseksi. Valikkojen ryh-



mittelyt ja toimintojen kulku on tarpeen kehittää niin selkeiksi kuin mahdollista. Tilanetta tosin vaikeuttaa se, että yhdelle ihmiselle selkeä valinta ei välttämättä ole lainkaan sitä toiselle, mutta tämä vaikeus on sama kaikessa suunnittelussa, ei pelkästään web-käyttöliittymissä.

Ylläpitoliittymä jakaantuu muutamaaan pääosastoon, jotka noudattavat enimmäkseen kohderyhmän ylläpitäjien sivustoillaan käyttämiä pääkategorioita ja tavallisimpia sisältötyyppejä. Nämä pääosastot muodostavat ylläpitoliittymän päävalikon. Niiden määrä on pieni, jotta valikosta ei tule liian laajaa: useita valikkokohteita sisältävästä valikosta on vaikeampi tehdä valinta kuin harvoja valikkokohteita sisältävästä, ja suurikokoinen valikko vie käyttöliittymästä enemmän tilaa kuin pieni. Käyttöliittymän käytettävissä oleva tila taas voi olla hyvinkin rajattu, sillä vaikka näyttöjen resoluutio onkin nykyisin useimmiten suurempi kuin 800 x 600 kuvapistettä, selainikkuna ei välttämättä ole yhtä suuri kuin näyttö eikä siltä voida sellaista edellyttääkään – osa ihmisistä, esimerkiksi minä, pitää mieluummin auki useita, hieman päällekkäisiä selainikkunoita, voidakseen esimerkiksi verrata kahdella eri sivulla olevaa sisältöä toisiinsa, tai ei muuten vain pidä erityisen leveistä selainikkunoista. On siis järkevää käyttää valikkoon niin vähän tilaa kuin mahdollista, jotta varsinaiselle sisällölle jää mahdollisimman paljon näyttöalaa.

Päävalikon sivustonhallinnan valikkokohteet ovat ylläpitotökalut, kategoriat, kuvat, linkit ja sisällöt. Nimet on pyritty valitsemaan siten, että käyttäjät ymmärtävät niiden merkityksen ilman erillisiä selityksiä. Näiden lisäksi on vielä uloskirjautumislinkki, sillä ylläpitoliittymään päästäkseen on tietenkin kirjauduttava sisään. Suunnittelun aikana kävin läpi erilaisia valikkokohdevaihtoehtoja, mutta tarkoituksenmukaisuuden nimissä päädyin lopulta mainittuun ratkaisuun. Pienemmälläkin linkkimäärällä olisi tässä kohdassa selvitty, mutta pyrin liittämään tärkeimmät käytössä olevat aiheet päävalikkoon, jotta yleisimmin käytettäviin toimintoihin pääsisi käsiksi mahdollisimman helposti. Toisaalta kohteita on hyvä voida muokata juuri siinä yhteydessä johon ne kuuluvat, joten kuvia, linkkejä ja sisältöjä voi muokata niin omissa osastoissaan kuin kategorioiden osinakin. Tämän on tarkoitus edesauttaa järjestelmän käyttömukavuutta, ennen kaikkea käytön joustavuutta.

Tarkempi selvitys käyttöliittymästä löytyy liitteestä 6.

### 4.2.3 Käytettävyydestestauksen suunnittelu

Tarkoitukseni oli toteuttaa järjestelmälle käytettävyydestaus, jossa kohderyhmästä koostettu testausryhmä kokeilisi käyttöliittymästä tehtyä mallia. En kuitenkaan saanut testausta suoritettua, kahdestakin syystä. Ensimmäinen syy oli yksinkertaisesti se, että tarkoitukseni oli suorittaa testaus demoversiolla, jossa sivusto olisi rakennettu dynaamisesti JavaScriptin avulla ylläpitoliittymässä syötetyillä tiedoilla, mutta käyttämäni koodustapa osoittautui liian kuormittavaksi ainakin senhetkisillä selaimilla. Olisin tietysti voinut muuttaa koodia, ja olisin epäilemättä niin tehnytkin, ellei toista syytä olisi ollut olemassa. Tämä oli nimittäin se, että en onnistunut kokoamaan testausryhmää, tai en ainakaan sellaista testausryhmää jonka olisin varmuudella saanut käsiini silloin, kun testaus olisi pitänyt suorittaa. Kohderyhmään kuuluvat henkilöt ovat usein hyvin kiireisiä omien projektiansa ja koulutöidensä kanssa, eikä niiltä, joihin sain yhteyden, liennyt aikaa testauksen suorittamiseen, ja testaus ilman testaaajia ei ole käytännössä erityisen hyödyllinen testaus.

Kohderyhmään kuulumattomista henkilöistä olisin pystynyt kokoamaan testausryhmän, mutta vaikka Krugin (2006, 134) mukaan testaus voidaankin sopivien henkilöiden puutteessa suorittaa myös henkilöillä, jotka eivät edusta ennakoitua käyttäjäkuntaa, siitä ei olisi mielestäni ollut tässä tilanteessa hyötyä. Kokoon saamani ryhmä olisi nimittäin koostunut ihmisistä, jotka eivät vastaa miltään osin nuorten harrastelijaylläpitäjien kohderyhmää: henkilöt eivät olisi nuoria, eivät harrastaisi visuaalisia asioita eikä heillä olisi mitään käsitystä tai kokemusta web-sivujen ylläpidosta saati käsitystä nuorten ylläpitäjien miellyttäviksi kokemista ulkoasuista, osaltaan ei edes minimaalista suurempaa käyttökokemusta tietokoneista. En siis usko, että tällainen ryhmä pystyisi sanomaan mitään merkittävää sen enempää järjestelmän ulkoasusta, rakenteesta kuin toiminnastaan, sillä järjestelmä on tarkoitettu nuorille ihmisille, jotka tietävät millaisia sivuja tahtovat ylläpitää, ja ainakin perusasioiltaan kuinka tietokoneita ylipäänsä käytetään. Sekä järjestelmän toiminnot että ulkoasu on suunniteltu tämän tavoitteen mukaan. Ryhmällä saataisiin selvyys siitä, kuinka hyvin järjestelmää pystyvät käyttämään ihmiset, joilla ei ole aavistustakaan siitä mitä he yrittävät tehdä, mutta mainitunlaisesta tutkimustuloksesta tuskin olisi hyötyä tässä yhteydessä.

Koska en saanut koottua sopivaa testausryhmää, päädyin siihen, että en suorita käytettävyydestä tämän projektin aikana. Päätös ei tietenkään tarkoita, että testausta ei olisi tarpeen suorittaa ennen lopullisen version valmistumista, mutta järjestelmään jää kuitenkin tämän projektin jälkeen vielä kehitettävää. Päätin, että käytettävyydestä voi suorittaa myöhemmässäkin vaiheessa, vaikka se olisikin hyödyksi jo suunnittelun alkuvaiheessa. Keskityin siis tässä projektissa rakentamaan käyttöliittymän ja toteutuksen omasta mielestäni tarkoituksenmukaisimmalla tavalla, sillä kuulun itse järjestelmän ensisijaisimpaan kohderyhmään, ja olen joka tapauksessa tutkinut muita vastaavia sivustoja ja niiden toimintaa, mikä voidaan Krugin (2006, 144) mukaan laskea testauksen ensimmäiseksi vaiheeksi. Tosin käyttöliittymän testaajan ei tulisi olla järjestelmän suunnittelijan tai kenenkään muunkaan sellaisen, joka on työskennellyt järjestelmän laatimisessa (Krug 2006, 133), mutta kaikkea ei voi saada. Vaikka en siis toteuttanutkaan käyttöliittymän testausta testiryhmällä, suunnittelin kuitenkin miten sen suorittaisin, joten esitän vain suunnittelutyön tulokset.

Käyttöliittymätestauksen kuvaus löytyy liitteestä 7.

### 4.3 Toinen järjestelmä

Koska ensimmäisen järjestelmän suunnitelma osoittautui umpikujaksi, ja koska päädyin siihen tulokseen että on huomattavasti käytännöllisempää tehdä itsenäisen järjestelmän sijaan lisäosa phpBB-foorumille kuten jo aiemmin on selostettu, hyllytin ensimmäisen järjestelmän suunnitelman. Toinen järjestelmä on siis phpBB-lisäosa, mikä asettaa sille tiettyjä rajoituksia, mutta antaa sille myös tiettyjä etuja, joista tärkeimpänä se, että osa vaadituista ominaisuuksista on olemassa jo valmiina, eikä niitä näin ollen tarvitse enää suunnitella eikä toteuttaa.

Tässä vaiheessa en enää suunnitellut järjestelmää sen enempää tietokannan kuin käyttöliittymänkään pohjalta, vaan päädyin suunnittelemaan yhtä toimintoa kerrallaan, sen jälkeen kun olin ensin koonnut itselleni listan niistä toiminnoista ja tiedoista, jotka järjestelmän tulisi ainakin sisältää. Päädyin siis toteuttamaan lähinnä prototyyppilähestymistapaa, varsinkin kun aloin samalla luoda järjestelmästä enemmän tai vähemmän toimivaa versiota. Toisen järjestelmän suunnittelun eteneminen poikkeaa näin ollen jonkin

verran ensimmäisen järjestelmän suunnittelun etenemisestä, ja ryhmittelen nyt suunnittelun etenemisen kuvauksen sen mukaisesti, kuinka olen sen tehnyt, yrittämättä kategorioida sitä jotenkin keinotekoisesti eri työvaiheiksi, joita siinä ei tosiasiaassa ole ollut.

Tarkempi kuvaus tietokannasta löytyy liitteestä 8 ja käyttöliittymästä liitteestä 9.

#### 4.3.1 Tyylien, käyttäjien ja oikeuksien hallinta

Nämä kolme toimintoa ovat osa varsinaista phpBB-ohjelmistoa, joten niitä ei oikeastaan tarvitse tässä suunnitella. Tyylien hallintaan riittää, että lisätään phpBB-ulkoasuille tarpeelliset mallinetiedostot, joiden avulla keskustelufoorumin ohjelmisto osaa rakentaa oikeat tyylit sen pohjalta, mitä varsinaisen php-koodisivun antamien komentojen mukaan sivulle tulee sisällyttää. Koodisivu siis kertoo mitä sivulla esitetään, mallinetiedosto kertoo sen miten se esitetään. Varsinainen tyylien asentaminen, muokkaus, poistaminen, ja käytössä olevan tyylin valinta tehdään siten, kuin se phpBB-foorumissa ylipäänsäkin tehdään, se ei siis edellytä toimenpiteitä tässä suunnitelmassa.

Vaikka käyttäjien ja käyttöoikeuksien hallinta ei kuulu työn rajaukseen, ne on kuitenkin sisällytetty tähän, koska ne on myös helppo soveltaa varsinaisesta phpBB-ohjelmistosta. Käyttäjien hallinta tulee itse asiassa keskustelufoorumista suoraan, sillehän voi rekisteröityä eri käyttäjiä, joille sitten määritetään eri oikeuksia, joiden perusteella käyttäjät voivat tehdä eri asioita. Käyttöoikeudet voidaan tallentaa suoraan phpBB:n omaan käyttöoikeustauluun, vaikkakin niiden saaminen toimintaan tarvittavalla tavalla vaatii pieniä mukautuksia käyttöoikeuksien hallintaa määrittävässä koodissa, mutta se taas liittyy enemmän järjestelmän toteutukseen kuin sen suunnitteluun eikä näin ollen ole varsinaisesti osa tätä työtä, joka sisältää vain suunnitteluosion, olkoonkin että tämän työn tekemisen puitteissa jo tulin kirjoittaneeksi tarvittavat muutokset ohjelmakoodiin niin, että tosiasiaassa olen jo suorittanut tämän vaiheen toteutuksenkin.

#### 4.3.2 Sivuston asetusten hallinta

Keskustelufoorumi phpBB sisältää jo sinällään osan sivuston tarvitsemista asetuksista, kuten esimerkiksi sen nimen ja käytössä olevan tyylin. Kaikkia sivuston asetuksia ei siis

tarvitse määrittää erikseen itse sisällönhallintajärjestelmässä. Sen sijaan tuntuu mielekkäältä ratkaisulta tallentaa asetukset samaan tapaan kuin keskustelufoorumi tallentaa omat asetuksensa. Itse asiassa asetukset voi tallentaa samaan tietokantaan kuin keskustelufoorumin omatkin asetukset, eikä niille näin ollen ole tarvetta luoda omaa tietokantatauluun. On kuitenkin asetuksia, jotka eivät sisälly phpBB:n omiin asetuksiin, joten näille on tarpeen luoda oma asetusten hallintasivu (kuva 5).

**ACMS**  
Täällä voit säätää ACMS:n yleisiä ja oletusasetuksia. Nämä asetukset vaikuttavat kaikkialla ACMS:ssä.

**Sivuston yleiset asetukset**

**ACMS käytössä**  
Tällä valinnalla voit ottaa ACMS:n käyttöön ja pois käytöstä. Mikäli järjestelmä on pois käytöstä, mikään ACMS:n osa ei näy käyttäjälle. Asetukset ja sisällöt eivät kuitenkaan katoa, ellei poista järjestelmän asennusta.

Kyllä  Ei

**Oletusasetukset**

**Näytä sisältöryhmät:**  
Jos tämä on käytössä, uusi sisältöryhmä oletusarvoisesti näytetään. Voit silti valita sisältöryhmän asetuksissa olla näyttämättä sitä. Huomaa, että tämä koskee vain uusia sisältöryhmiä, olemassaolevien arvot eivät muutu.

Kyllä  Ei

**Lisää uudet sisältötyypit valikkoon:**  
Jos tämä on käytössä, uusi sisältötyyppi lisätään oletusarvoisesti valikkoon. Voit valita sisältötyypin kohdalla lisätäanko se valikkoon vai ei, tämä vain määrittää oletusarvon. Huomaa, että tämä koskee vain uusia sisältötyyppejä, olemassaolevien arvot eivät muutu.

Kyllä  Ei

**Sisältötyyppiä sivulla:**  
Tämä määrittää oletusarvon sille, kuinka monta sisältötyypin sisältöä näytetään yhdellä sivulla. 0 tarkoittaa, että kaikki sisällöt näytetään. Arvon voi ohittaa sisältötyypin asetuksista.

**Submit changes**

Kuva 5. ACMS:n yleisten asetusten hallintasivu. Kuvassa on vain sivun periaate, siinä ei ole näkyvissä kaikkia asetuksia.

Ensimmäisen järjestelmän asetusten suhteen päädyin ratkaisuun, jossa sivuston asetukset tallennetaan omaan tietokantatauluunsa siten, että jokaisella asetuksella on taulussa oma sarakkeensa. Tämä ratkaisu ei kuitenkaan ole enää ajankohtainen toisessa järjestelmässä, jossa käytetään phpBB:n omaa asetustietokantatauluun, koska olemassa oleva tietokantataulu sisältää vain kaksi saraketta, ”nimi” ja ”arvo”, joista sarakkeeseen ”nimi” sijoitetaan asetuksen nimi, ja sarakkeeseen ”arvo” sen arvo.

Luettelo toisen järjestelmän sivustoasetuksista on nähtävissä taulukossa 5.

Taulukko 5. ACMS:n yleiset asetukset. Nämä asetukset määrittävät ACMS:n oletusasetuksia järjestelmän muilla sivuilla, minkä lisäksi järjestelmä voidaan ottaa käyttöön ja pois käytöstä (jolloin käytetään pelkkää phpBB:ta) asetuksella ”ACMS käytössä”.

<b>Nimi</b>	<b>Oletus</b>	<b>Kuvaus</b>
ACMS käytössä	kyllä	Määrittää, onko ACMS käytössä vai ei.
Etusivu	index.php	Määrittää, mikä sivu toimii sivuston etusivuna (eli mikä sivu aukeaa kävijälle, joka tulee sivustolle määrittämättä mille sivuston sivulle tahtoo tulla).
Näytä sisältöryhmät	kyllä	Määrittää, näytetäänkö uudet sisältöryhmät oletusarvoisesti vai ei.
Lisää uudet sisältötyypit valikkoon	kyllä	Määrittää, luodaanko uusille sisältötyypeille oletusarvoisesti valikkoelementti vai ei.
Sisältötyyppejä sivulla	15	Määrittää, kuinka monta sisältötyyppiin kuuluvaa sisältöä näytetään oletusarvoisesti samalla sivulla ilman sivutusta.
Oletuskenttä	Teksti	Määrittää, mikä sisältökentän tyyppi on oletusarvoisesti valittuna, kun uutta sisältökenttää luodaan.
Tekstin näytettävät merkit	0	Määrittää, kuinka monta merkkiä pitkässä tekstissä näytetään oletusarvoisesti ennen Lue lisää -linkkiä, mikäli toiminto on käytössä.
Salli paikalliset kuvat	kyllä	Määrittää, sallitaanko kuvaelementin oletusarvoisesti käyttää kuvia foorumin kuvahakemistoon talletettuja kuvia.
Salli linkitettyt kuvat	kyllä	Määrittää, sallitaanko kuvaelementin oletusarvoisesti käyttää linkitettyjä kuvia.
Valintaelementin tyyppi	Pudotusvalikko	Määrittää, mikä on valintaelementin ole-

		tustyyppi valintaelementtiä luotaessa.
Ryhmän tyyppi	Sisältölista	Määrittää, mikä on ryhmäelementin oletustyyppi ryhmäelementtiä luotaessa.
Järjestä ryhmä automaattisesti	kyllä	Määrittää, järjestetäänkö ryhmäelementti automaattisesti vai ei.
Järjestystapa	Aikajärjestys	Määrittää, mikä on ryhmäelementin automaattisen järjestyksen järjestystapa.
Järjestyksen suunta	Nouseva	Määrittää, miten päin ryhmäelementti automaattisesti järjestetään.
Linkki uuteen ikkunaan	Ei	Määrittää, avataanko linkkielementin linkit oletusarvoisesti uuteen ikkunaan.
Ajan tyyppi	Ajankohta	Määrittää, onko aikaelementin aikatyypiksi oletusarvoisesti ajankohta vai kesto.
Moduulin tyyppi	Pudotusvalikko	Määrittää, millä tavalla moduulielementti oletusarvoisesti esitetään.
Moduulisältöjen määrä	5	Määrittää, kuinka monta moduulisältöä oletusarvoisesti moduulissa esitetään.

#### 4.3.3 Lokitietojen keruu ja esittäminen

Lokitietojen keruussa voidaan pääosin turvautua phpBB:n valmiisiin mekanismeihin, sillä ohjelmistossa on varsin hyvät lokitietojen keruu- ja esitysmahdollisuudet. Tietoja kerätään ylläpidollisista muutoksista kuten keskustelualueiden luonnista, muokkauksesta ja poistamisesta, tietyistä käyttäjäprofiilien muutoksista, ynnä muusta. Tietoja kerätään myös viestien, keskusteluketjujen, käyttäjien ja paikalla olevien kävijöiden määräästä ynnä muista vastaavista asioista.

Varsinaista kävijälaskuria, jota sivusto kaipaa, ei phpBB kuitenkaan tarjoa, joten se on tehtävä joko erillisenä lisäosana tai sisällönhallintajärjestelmän osaksi. Lisäksi on sivustolle käytännöllistä pitää kirjaa erityyppisten rakenteiden, esimerkiksi galleriakuvien,

määrästä. Näiden tietojen tallentamiseen voidaan kuitenkin käyttää foorumiohjelmiston valmiita tietokantatauluja sen sijaan, että niille pitäisi luoda omat taulut. Varsinainen tietojen tallentaminen ja esittäminen kuuluu jälleen kerran toteutusvaiheen suorituksiin, mutta tarvittavat tiedot on kirjattu taulukkoon 6.

Taulukko 6. ACMS:n lisäykset foorumin lokitietoihin ja asetuksiin. ”Kävijäloki” on tätä tarkoitusta varten luotava uusi taulu, johon lisätään taulukossa esitetyt kolme kenttää. ”Sisältöjen loki” on uusi lokityyppi, joka syötetään phpBB:n lokitauluun. Lokiin merkittävät arvot sisältävät taulukossa esitetyt tiedot. ”Foorumin tilastot” on phpBB:n tilastotietoja säilyttävä taulu, johon lisätään merkinnät taulukossa esitetyille arvoille.

Nimi	Merkintä
Sisältöjen loki	Sisällön nimi
	Sisältö lisätty/muokattu/poistettu
Kävijäloki	Kävijän IP-osoite
	Ladattu sivu
	Ajankohta
Foorumin tilastot	Uniikit kävijät (sama IP-osoite lasketaan enintään kerran vuorokaudessa)
	Sivulataukset (jokainen kerta kun sivu ladataan, lasketaan tähän lukuun)

#### 4.3.4 Tietokokonaisuuksien hallinta

Tietokokonaisuuksien hallinta on olennainen osa sivuston rakenteen hallintaa, tai paremminkin sanottuna, tietokokonaisuuksien hallinta on koko sivuston rakenteen hallinnan pohja. Jokainen osio, joka sivustolla on, on yksi tietokokonaisuus, joka sitten jakautuu pienemmiksi tietokokonaisuuksiksi. Esimerkki sivuston rakenteen jakamisesta tietokokonaisuuksiksi on taulukossa 7.



Taulukko 7. Yksinkertainen sivuston rakenne jaettuna tietokokonaisuuksiksi.

Kategoria	Sivu	Osa	Osan osa	
Sivusto	Etusivu	Otsikko		
		Tervetuloteksti		
		Uusin päivitys		
		Päivityssivun linkki		
	Päivitykset	Päivitysluettelo		Päivitys
				Ilmoitus
	Vieraskirja	Ohjeteksti		
		Viestiluettelo		Viesti
				Vastaus
		Viestin kirjoitus		
Linkit	Linkkiryhmä		Linkki	
Sarjakuva	Arkisto	Sarjakuvan nimi		
		Sarjakuvan luku	Luvun nimi	
			Luvun sivut	
	Lukusivu	Sarjakuvasivu		
Sivun numero				
Tekstit	Arkisto	Tekstiluettelo	Tekstin nimi	
			Linkki tekstiin	
			Kuvaus	
	Lukusivu	Tekstin nimi		
Varsinainen teksti				
Kuvat	Galleriasivu	Aihepiiri	Kuvat	
	Kuvasisu	Kuva		
		Kuvan tiedot	Nimi	

			Kuvaus
			Valmistumisaika

Suunnittelun pohjaksi soveltuu mainiosti phpBB:n oma keskustelualueiden hallinta, jolla käytännössä määritellään keskustelufoorumin rakenne, ja moduulien hallinta, jolla määritellään foorumin käytettävissä olevat osat ja valikkojen, esimerkiksi käyttäjän omista asetuksissa olevien, rakenne. Sisällönhallintajärjestelmässä ei tietenkään määritellä keskustelualueita, mutta periaate on sivuston rakenteen määrittämiseenkin varsin toimiva. Jotkin osiot ovat päätasolla, toiset niiden alaosioina, oli kyseessä sitten keskustelualue ja sen sisältämät toiset keskustelualueet tai sivuston päätason linkki ja sen alla olevat alemman tason linkit.

Keskustelualueen ja tietokokonaisuuden määrittämiseen luonnollisesti tarvitaan kuitenkin eri tiedot, ja tietokokonaisuuden määrittämisessä niitä tarvitaan enemmän. Näin ollen tässä tilanteessa ei ole mahdollista käyttää phpBB:n valmiita tauluja, vaan niitä varren on luotava uudet. Periaatteeltaan nämä tietokokonaisuudet ovat samoja kuin ensimmäisen järjestelmän kategoriat, mutta voidakseen toimia saumattomasti phpBB-foorummin kanssa ne vaativat hieman enemmän tietoja tallennettavaksi tietokantaan ja näytettäväksi hallintasivulla. Päästäkseni selville siitä, mitkä tiedot oikeastaan olisivat tarpeen jotta toivottu toiminnallisuus saavutettaisiin, testasin asiaa foorumilla. Luettelo tarpeellisiksi havaituista tiedoista löytyy taulukosta 8.

Taulukko 8. Kategorioiden hallintasivulla tarvittavat tiedot. Samoilla tiedoilla hallitaan myös sivuja ja sisältötyyppejä (rakennepohjia, nimi muuttui suunnittelun kuluessa ja toisen järjestelmän käyttöliittymässä käytetään nimeä ”sisältötyyppi” eikä ”rakennepohja”), mutta kerrottakoon ne nyt tässä yhteydessä.

Nimi	Kuvaus	Lisähuomioita
Sisältötyypin tyyppi	Määrittää, minkä tyyppinen sisältötyyppi on kyseessä.	Voi olla kategoria, sivu, sisältötyyppi tai moduuli.
Sisältöryhmä	Määrittää, mihin ryhmään sisältötyyppi kuuluu.	Ryhmä on sisältötyyppien valinnainen luokittelutapa.

Isäntätyyppi	Määrittää, mihin ylemmän tason tyyppiin sisältötyyppi kuuluu.	Jos isäntätyyppiä ei ole, sisältötyyppi on ylätason tyyppi.
Sisältötyypin nimi	Sisältötyypin nimi ylläpitoliittymässä.	Kävijälle näytetään eri nimi, joka määritetään kielikohtaisesti.
Sisältötyypin kuvaus	Sisältötyypin kuvaus ylläpitoliittymässä.	Kävijälle näytetään eri kuvaus, joka määritetään kielikohtaisesti.
Sisältötyypin kieli	Kieli, jolla sisältötyyppi näytetään.	Tämä mahdollistaa eri sisällön näyttämisen erikielisille käyttäjille, jotta esimerkiksi englanninkielisten käyttäjien ei tarvitse ihmetellä suomenkielisiä tekstejä.
Näytä valikossa	Määrittää, käytetäänkö sisältötyyppejä sivuston valikkojen rakentamiseen.	Tämä toiminto lisää automaattisesti sisältötyypin sivuston päävalikkoon kohtaan, johon se määrittystensä puolesta loogisesti kuuluu. Valikkorakennetta voi muuttaa eri valikoissa niin, että sisältötyypin voi halutesaan näyttää muuallakin.
Sisältöjä sivulla	Tämä asetus määrittää, kuinka monta sisältötyypin sisältöä näytetään ilman sivutusta.	Jos arvo on asetettu, se ohittaa sivustoisten sisältöjen määrän oletusasetuksen.

Voi kuitenkin olla tarpeen pystyä luokittelemaan samat tietokokonaisuudet useammalla kuin yhdellä tavalla. Ylläpitäjä voi tahtoa kävijöiden näkevän sisällöt aihepiireittäin, mutta tahtoo itse pystyä hallinnoimaan niitä tyypeittäin, esimerkiksi kaikki galleriat yhdessä ryhmässä ja kaikki tekstit toisessa ryhmässä. Koska yksi tietokokonaisuus voi siis kuulua yhteen tai useampaan, tarkemmin määrittelemättömään määrään muita kokonaisuuksia, ei tuntunut järkevältä tallentaa tätä tietoa samaan tietokantatauluun muiden tietokokonaisuutta koskevien tietojen kanssa – tällöinhän jälleen olisi joko rajoitettava ylläpitäjän valinnanvapautta tai luotava järjestelmä, joka luo uusia kenttiä tietokantatauluun.

luun tarpeen mukaan, ja koska näitä kumpaakin yrittää järjestelmä muutenkin välttää, eivät ne vaikuttaneet järkeviltä vaihtoehdoilta tälläkään kertaa.

Ratkaisuksi ongelmaan päädyin luomaan toisen tietokantataulun, johon tallennetaan tieto näistä tietokokonaisuuksien ryhmittelyistä, joita kutsun nimellä ”ryhmät”, sillä ryhmiähän ne käytännössä joka tapauksessa ovat. Päädyin siihen, että sivuston rakenne määritellään tietokokonaisuuksien hallinnassa, ja ryhmien avulla luokitellaan tietokokonaisuuksia erilaisiksi kokonaisuuksiksi muun tarpeen mukaan. Erityinen valinta on ryhmään lisätty sille, näytetäänkö niitä muualla kuin ylläpidon asetuksissa: nämä ryhmät käyttäjä voi nähdä phpBB:n omissa asetuksissaan, ja mahdollisesti luoda tai ainakin selata omia sisältöjään näissä ryhmissä, mikäli hänen käyttöoikeutensa sen sallivat. Ensimmäisessä sijaisesta tämä on tarkoitettu sitä varten, että käyttäjä voi nähdä kommenttinsa luokiteltuina sisällön tyyppin mukaisesti, esimerkiksi yhdessä listassa kuvien, toisessa tekstien kommentit, kuuluuhan kommentointimahdollisuus osaltaan web-sisällönhallintajärjestelmien ominaisuuksiin, ainakin uutisointien kohdalla. Ryhmien ominaisuudet on kirjattu taulukkoon 9.

Taulukko 9. Sisältöryhmän hallintasivulla tarvittavat tiedot.

<b>Nimi</b>	<b>Kuvaus</b>	<b>Lisähuomioita</b>
Sisältöryhmän nimi	Sisältöryhmän nimi ylläpitoliittymässä.	Kävijälle näytetään eri nimi, joka määritetään kielikohtaisesti.
Sisältöryhmän kuvaus	Sisältöryhmän kuvaus ylläpitoliittymässä.	Kävijälle näytetään eri kuvaus, joka määritetään kielikohtaisesti.
Sisältöryhmä näytetään	Määrittää, näytetäänkö ryhmä käyttäjän hallintapaneelissa.	Tällä voi tehdä piilotettuja ryhmiä, joiden avulla voi luokitella asioita ylläpitoliittymässä ilman että luokittelu näkyy käyttäjille.

Sisältöjen hallinta ei myöskään ole täysin erillään tietokokonaisuuksien hallinnasta, sillä phpBB 3 -foorumeilla keskusteluketjuja sisältävät keskustelualueet ja pelkästään muita keskustelualueita sisältävät kategoriat ovat vain saman asian kaksi eri muotoa ja tallennetaan samaan tietokantaan. Koska järjestelmäni on tarkoitettu lisäosaksi juuri tällaiselle

foorumille, olen ottanut saman periaatteen myös sivurakenteiden ja sivun rakennepohjien tallentamiseen. Näin ollen sisältöjen hallinta on oikeastaan myös osa tietokokonaisuuksien hallintaa ja päinvastoin, mutta lukemisen selkeyttämiseksi ne on tässä raportissa kuitenkin jaoteltu erillisiin osioihin, joista rakennepohjien hallinta kuuluu tietokokonaisuuksien hallintaan ja itse sisällöt (tekstit, uutiset, kuvat ja niin edespäin) kuuluvat sisältöjen hallintaan. Koska myös tiedot on samalla periaatteella tallennettu eri tauluihin tietokannassa, jako vaikutti loogiselta.

Rakennepohjien tietokantaan tallentamisen suunnittelusta on selostettu tarkemmin ensimmäisen järjestelmän suunnittelun selostuksessa, ja koska toisessa järjestelmässä asia on käytännössä rakennettu ensimmäisen järjestelmän pohjalle, en selosta tässä sen enempiä sitä, miten tapa tallentaa tiedot muodostui, sillä kaikki ensimmäisen järjestelmän kohdalla tästä asiasta sanottu pitää paikkansa myös toisen järjestelmän kohdalla. Totean siis vain, että rakennepohjat voidaan tallentaa samaan tietokantatauluun kuin tietokokonaisuuksien määrittelyyn tarkoitettujen rakenteidenkin, ja että tosiasiansa rakennepohjat ja rakennepohjien kokonaisuudet ovat vain kaksi eri versiota samasta asiasta, samoin kuin keskustelufoorumien keskustelualueet ja keskustelufoorumien keskustelualueiden kokoelmat. Niissä on tietenkin toiminnallisia eroja, muutenhan näitä kahta ei edes kannattaisi erotella, mutta tämä toteutetaan pääsääntöisesti ohjelmallisesti. Tärkeimpänä tietokantaeroavuutena on se, että rakennepohjien on sisällettävä tietokenttiä, kun taas kategoriat sisältävät rakennepohjia tai toisia kategorioita.

Koska phpBB on monikielinen ympäristö, halusin myös järjestelmän jollain tapaa tukevan monikielisyyttä. Vaikka phpBB:n keskustelualueissa ei olekaan mahdollisuutta valita, että tietyt alueet näytettäisiin vain tietyillä kielillä (tai jos on, en ole ainakaan löytänyt sitä), tahdoin omassa järjestelmässäni tällaisen mahdollisuuden, jotta ylläpitäjä voi halutessaan tehdä sivustostaan versiot eri kielille. Joitakin monikielisiä sivustoja olen kohderyhmän tuotoksissa nähnyt, ja lisäosana keskustelufoorumille järjestelmästä todennäköisesti voisi tulla käyttökelpoinen laajemmallekin ihmisryhmälle, jolloin myös tuki monikielisyydelle voi olla tärkeää, varsinkin jos keskustelufoorumille on asennettu useampi kuin yksi kieli. Tällöin voidaan tehdä eri kielille eri versiot sivustosta helposti.

Koska keskustelufoorumille asennettujen kielten määrä voi olla mitä tahansa yhdestä äärettömään, päädyin siihen, että kukin rakennepohja voidaan asettaa toimimaan yhdellä

tai useammalla kielellä, ja että tarvitaan erillinen tietokantataulu rakennepohjan käännöstä vaativien tekstien tallentamiseen näillä kielillä. Toinen vaihtoehto olisi ollut sama kuin phpBB:n keskustelualueiden kohdalla, eli se, että kaikki rakennepohjat näytetään kaikille kieliversioille, mitä en halunnut; kolmas vaihtoehto olisi ollut se, että jokainen rakennepohja on mahdollista esittää vain yhdellä kielellä, ja vaikka se olisi ollutkin toteutukseltaan helpompi, olisi se ollut käytöltään vaikeampaa, sillä jos sama rakenne halutaan esittää eri kieliversioissa, se täytyisi tällöin tehdä ja tallentaa erikseen jokaiselle kieliversiolle sen sijaan että käytettäisiin samoja tietoja ja asetuksia kaikissa kieliversioissa ja ainoastaan muutetaan käännökset tietyille kohdille.

#### 4.3.5 Sisältöjen hallinta

Sisältöjen hallinta pitää sisällään toiminnot ”Tekstisisällön hallinta”, ”Uutisten hallinta” ja ”Kuvien hallinta”, sillä tarkemman tutkimus- ja suunnittelutyön jälkeen järjestelmä päätyi muotoon, jossa nämä kaikki kolme toimintoa voidaan suorittaa samalla tavalla. Ei siis ole syytä suunnitella niitä erikseen, koska lopputuloksena olisi vain kolme suunnitelmaa, joiden ainoa ero on siinä mitä termiä niistä käytetään.

Tekstisisällöiksi voi nuorten harrastelijaylläpitäjien sivustoilla määritellä ainakin erilaiset kirjoitelmat ja taustatiedot tarinoista, joita voi olla sivustoilla vaihteleva määrä. Lisäksi tekstisisältöjä voi olla muunkinlaisia, mutta yleensä niissä kerrotaan kuitenkin jotakin jostakin aiheesta. Mainittu kuvaus on itsestäänselvyydestään huolimatta varsin oivallinen tapa määritellä sitä, minkälainen on nuoren harrastelijaylläpitäjän sivuston tekstisisältö, sillä ne todellakin voivat olla eri sivustoilla, ja myöskin saman sivuston eri osissa, hyvinkin erilaisia, eikä mitään yhteistä määritelmää välttämättä voi edes tehdä.

Uutisina toimivat lähinnä erilaiset sivuston päivityksestä kertovat ilmoitukset ynnä muut mahdolliset tiedotteet, joita voi sivustosta riippuen olla joko paljon, vähän tai ei lainkaan. Yleensä sivustolla on kuitenkin nähtävissä vähintäänkin viimeisimmän päivityksen tiedot, vaikka niitä ei olisikaan arkistoitu minnekään sen enempää. Kyseessä on kuitenkin aina jonkinasteinen tiedonanto sivuston selaajille, joten sen voi hyvin määritellä nimikkeellä ”uutinen”, onhan se selaajalle uutta, ja vaikka se ei olisikaan sinänsä

uutta, se on periaatteeltaan kuitenkin hyvin lähellä sitä, mitä esimerkiksi Joomla! kutsuu nimellä ”uutinen” (vertailu kuvassa 6).

## Uutinen

Category: [Joomla!](#)  
 Published on Thursday, 18 December 2014 00:12  
 Written by [REDACTED]  
 Hits: 0

Tämä on uutinen.

[< Prev](#)

**24.12.2013**

Sivuston uusi ulkoasu julkaistu. Hyvää joulua!

Kuva 6. Joomla!n uutinen (ylempi, valkoisella pohjalla) ja tyypillinen päivitystiedote (alempi, vaaleansinisellä pohjalla). Joomla!n uutisessa on enemmän tietoja, mutta periaate on yhtä kaikki hyvin samankaltainen.

Kuvat ovat useimmiten tärkeässä osassa kohderyhmän sivustoilla, ja miltei jokaisella onkin yksi tai useampi kuvagalleria, jossa esitellään ylläpitäjän tuotoksia. Ainoat poikkeukset kuva-asiassa ovat sivustot, joiden tarkoituksena on esitellä kirjoitettuja tarinoita. Näilläkin sivustoilla saattaa olla kuvamateriaalia, mutta se ei ole välttämättömyys. Joillakin sarjakuvan julkaisemiseen keskittyneillä sivustoilla ei ole kuvagalleriaa, mutta koska sarjakuva itsessään on teknisesti ottaen kuva, vaikka se voikin sisältää myös tekstiä, kuvien hallinta pätee myös näillä sivustoilla.

Kuten todettua, ei kuitenkaan ole tarpeen jaotella näitä kolmea erikseen, sillä niitä voidaan kaikkia kutsua yhteisnimellä sisällöt. Ne ovat pohjimmiltaan jopa niin samanlaisia keskenään, että ne voidaan tallentaa keskenään samaan tietokantatauluun, vaikka tämän asian toteaminen vaatikin pidempää keskittymistä ja pintapuolista suurempaa perehtymistä asiaan. Kun tiedot jaetaan riittävän pieniin osiin, saadaan paitsi kaikki luetellut sisällöt jaettua yhteneväisiin tallennettaviin palasiin, myös järjestelmä, jolla voidaan tallentaa milteipä minkä muotoinen sisältö tahansa. Analyysi tästä löytyy taulukosta 10.

Taulukko 10. Näytesivustojen päärakenteiden jako pienempiin osiin.

Rakenne	Osa	Tyyppi
Galleria	Kuvaluokka	ryhmäelementti
		tekstirivi (ryhmän otsikko)
Kuvaluokka	Kuva	URL-osoite (esikatselukuva)
		URL-osoite (katselusivu)
		tekstirivi (kuvan vaihtoehtoinen teksti)
Kuva	Kuvatiedosto	URL-osoite
	Nimi	tekstirivi
	Valmistumisaika	ajankohta/päivämäärä
	Oikeudet	luettelo
	Kuvaus	tekstikappaleita
Tarina	Luku	ryhmäelementti
		tekstirivi (ryhmän otsikko)
Luku	Nimi	tekstirivi
	Lukusivu	URL-osoite
	Sivunumero	kokonaisluku
Sarjakuva	Sarjakuva	URL-osoite
	Sivunumero	kokonaisluku
Teksti	Nimi	tekstirivi
	Kuvaus	tekstikappale
	Teksti	tekstikappaleita
Linkkilista	Linkkiryhmä	ryhmäelementti
		tekstirivi (ryhmän nimi)
Linkkiryhmä	Linkki	URL-osoite (linkki)
		URL-osoite (banneri)

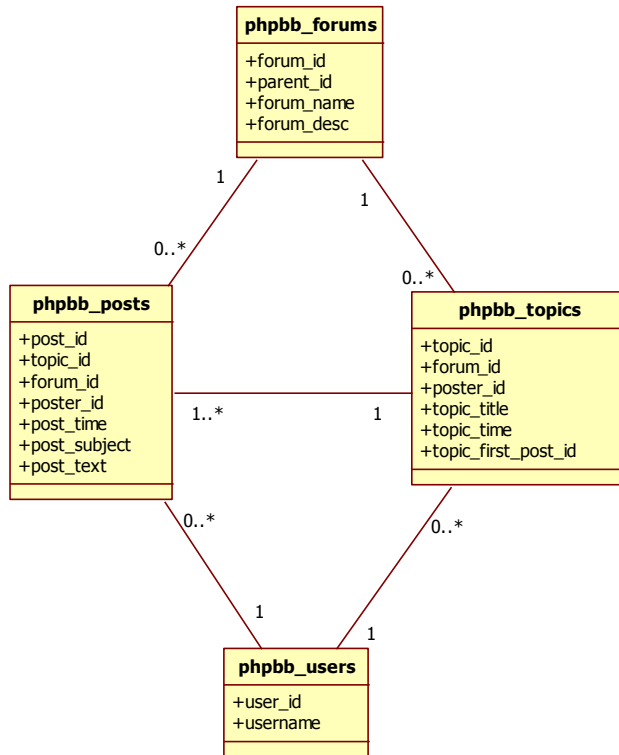


		tekstirivi (linkin nimi)
Päivitysluettelo	Päivitys	ajankohta/päivämäärä
		luettelo (päivitystapahtumista)
		tekstikappaleita

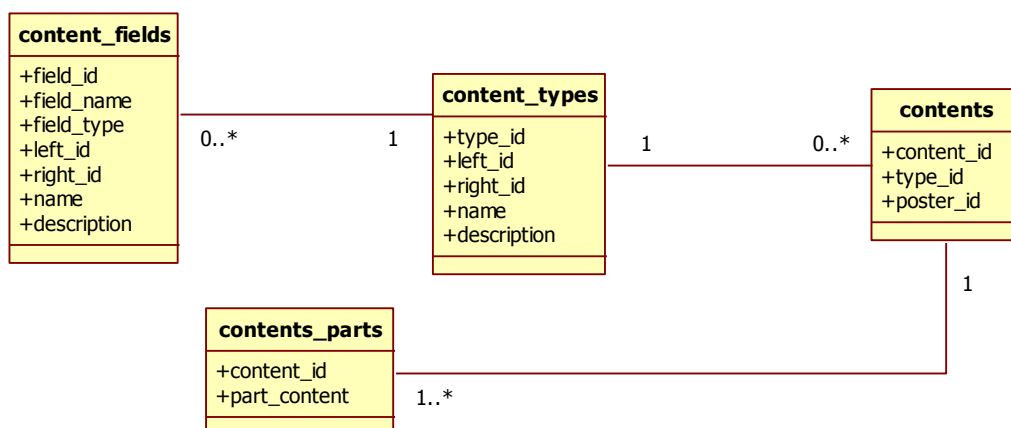
Varsinainen sisältöjen jako osiin kuuluu kuitenkin tietokokonaisuuksien hallintaan, sillä rakennepohjat luodaan siellä. Sisältöjen hallintaan sen sijaan kuuluu itse sisältöjen lisääminen, muokkaaminen ja poistaminen, siis se, mitä sisällönhallintajärjestelmällä varsinaisesti tehdään: tietokokonaisuuksissa luodaan rakenteet, mutta rakenteista ei ole paljoakaan hyötyä ilman sisältöä. Tietokokonaisuuden määrittäminen kertoo, missä muodossa tieto esitetään, mutta se, mitä tietoa esitetään, määritetään sisällöissä.

Alkuperäisen suunnitelman mukaan sisällöt olisi tallennettu oman tyyppinsä mukaisiin tietokantatauluihin. Tässä mallissa ei olisi ollut erikseen rakenteen määrittelyä, vaan sisällön rakenne olisi ollut sama kuin tietokantataulun rakenne. Joustavamman järjestelmän vuoksi kuitenkin päädyin määrittämään sisällön rakenteen erillisessä tietokantataulussa, mistä seurasi tarve paitsi jakaa sisällöt pienempiin osiin määrittämisen mahdollistamiseksi, myös tarve tallentaa ne näissä pienemmissä osissa, samaan tapaan kuin keskustelufoorumien viestiketjut.

Viestiketjut on, ainakin phpBB-foorumeissa, tallennettu kahteen eri tietokantatauluun: yhdessä taulussa on tallennettuna tiedot viestiketjuista itsestään, toisessa taulussa ovat viestit (kuva 7). Tähän samaan tapaan siis päädyin tallentamaan sisällöt, olihan keskustelufoorumissa jo asiasta hyvä malli. Yhteen tietokantatauluun siis tallennetaan perustiedot sisällöstä, siis sen nimi ja muut vastaavat tiedot, toiseen tietokantatauluun ne pienemmät osat joihin sen rakenne on jaettu, eli tietokentät tietorakenteiden määrittelystä. Esimerkiksi galleriakuvien kohdalla yksi tieto olisi kuvan osoite, toinen tieto olisi sen kuvaus, ja kolmas olisi sen valmistumispäivämäärä, kun taas sisällön määrittävään tauluun tallennettaisiin kuvan nimi ja sisällön tyyppi (kuva 8).



Kuva 7. phpBB-foorumien viestiketjurakenne. Kuvassa on yksinkertaistettu versio, todellisissa tietokantatauluissa on huomattavasti enemmän kenttiä. Viestiketjut määritetään yhdessä taulussa, viestiketjujen viestit löytyvät toisesta. Kummastakin taulusta on viittaus users- ja forums- tauluihin, koska jokaisella viestillä ja viestiketjulla on kirjoittaja ja ne kuuluvat jollekin keskustelualueelle.



Kuva 8. ACMS:n sisältöjäsentelusrakenne, yksinkertaistettu versio. Tieto sisällöistä tallennetaan contents-tauluun. Tieto siitä, mikä oikeastaan on sisällön rakenne, määritetään content\_types-taulussa (sisällön tyyppi) ja content\_fields-tauluissa (sisällön tieto-

kentät ja niiden järjestys). Tämän informaation pohjalta osataan rakentaa varsinainen sisältö, jonka kenttien sisältö on tallennettu contents\_parts-tauluun.

#### 4.3.6 Omien tietokenttien hallinta

Omien tietokenttien hallinta on oikeastaan osa tietokokonaisuuksien hallintaa, sillä kuten jo ensimmäisen järjestelmän suunnittelun yhteydessä mainittiin, käyttäjien omat ja valmiiksi tarjottavat rakennepohjat tallennetaan samalla tavalla. Toisessa järjestelmässä ei välttämättä oikeastaan voi edes tehdä minkäänlaista eroa näiden kahden välillä, ellei sitten sillä tavalla, että ylläpitäjälle voidaan asennuksen yhteydessä tarjota mahdollisuus asentaa oletusrakennepohjat tai vaihtoehtoisesti mahdollisuus olla asentamatta niitä. Oli miten oli, rakennepohjia ja niiden tietokenttiä voi siis lisätä, poistaa ja muokata mielensä mukaan, mikäli käyttäjällä on riittävät käyttöoikeudet järjestelmässä.

Erillisten tietokenttien olemassaolo ylipäänsä on seurausta siitä, että rakenne on tehty joustavaksi. Jotta rakennepohjista voidaan tehdä mahdollisimman muokattavia ilman, että asetetaan rajoituksia sille kuinka monta tietokenttää rakennepohjassa voi kaikkiaan olla, on ainoa vaihtoehto määritellä tietokentät erillisessä tietokantataulussa, josta ne sitten linkitetään itse rakennepohjaan. Analysoituani sivustoja tarkemmin ja jaettuani sivustojen rakenteita pienempiin osiin päädyin luetteloön osista, joiden avulla voidaan rakentaa mikä tahansa näistä sivustoista. Tutkailtuani tätä luetteloa huomasin, että joukossa ovat oikeastaan miltei kaikki HTML-kielen rakenteet, joten päädyin täydentämään luettelon niillä rakenteilla, jotka luettelosta vielä puuttuivat, jotta ylläpitäjä voisi luoda sivustonsa myös näitä HTML-rakenteita käyttäen (taulukko 11).

Taulukko 11. Sivustorakenteen kenttävaihtoehdot ja niiden vastaavuus HTML-koodissa.

<b>Kenttätyyppi</b>	<b>HTML-vastine</b>
Tekstirivi	span, em, p, h1, h2, h3, h4, h5, h6 (tekstit)
Teksti	p (tekstikappale)
Kuva	img (kuva)
Valinta	radio, checkbox, select (valinnat)

Ryhmä	ul, ol, dl (listat), table (taulukko)
Linkki	a (linkki)
Aika	(ei tarkkaa vastinetta)
Moduuli	(ei tarkkaa vastinetta)

Tällaisenaan luettelo on kuitenkin vain lista sivustossa esiintyvistä kentistä, ja rakennepohjan rakentamiseen tarvitaan myös määrittämiä siitä, kuinka kenttä esitetään, miten se liittyy rakennepohjaan ja miten se suhtautuu toisiin kenttiin. Täydensin siis luetteloita tarvittavilla lisätiedoilla ja kentän lähetykseen tarvittavilla tiedoilla, sillä sisällöt lisätään sivustolle täyttämällä tiedot lomakkeessa, jonka rakenne on määritetty tallennetun rakennepohjan asetuksissa. Lomakkeen kentät on siis määritetty juuri näiden omien kenttien avulla, ja kentän asetuksissa on säädettävä, minkä muotoinen on sisältö jota sillä lähetetään. Täydennetty, tarvittavan toiminnallisuuden mahdollistava luettelo on taulukossa 12. Tämän taulukon mukaisesti tietokenttien tiedot tallennetaan omaan tietokanta-tauluunsa.

Taulukko 12. ACMS:n tietokenttätyyppit ja niiden ominaisuudet.

Kenttätyyppi	Ominaisuus	Kuvaus
Kaikille tyypeille yhteiset	Kentän tyyppi	Määrittää, mikä on kentän sisällön tyyppi.
	Kentän nimi	Kentän nimi ylläpidon hallintapaneelissa tai sisältötyyppiin sisältöä lähetettäessä.
	Kentän kuvaus	Kuvaus kentästä hallintapaneelissa tai sisältötyyppiin sisältöä lähetettäessä.
	Pakollinen kenttä	Määrittää, onko kentässä oltava arvo.
	Valvottava kenttä	Määrittää, tuleeko valvojan hyväksyä tähän kenttään tehtävät muutokset.
Tekstirivi	Tekstirivin tyyppi	Määrittää, onko kyseessä numeroarvo vai merkkijono (joka voi tietysti olla yhtä lailla luku kuin tekstikin).

	Tekstirivin pituus	Määrittää tekstilaatikon koon sisältöä lähetettäessä (ei pakollinen).
	Tekstirivin maksimikoko	Määrittää merkkijonon suurimman sallitun pituuden (ei pakollinen).
	Salli BBCode	Määrittää, saako tekstissä käyttää phpBB:n muotoilukomentoja (BBCode).
	Oletusarvo	Määrittää tekstikentän oletusarvon.
Teksti	Rivit	Määrittää tekstilaatikon korkeuden sisältöä lähetettäessä (ei pakollinen).
	Merkkejä rivillä	Määrittää tekstilaatikon leveyden sisältöä lähetettäessä (ei pakollinen).
	Minimipituus	Määrittää tekstin pienimmän sallitun pituuden (jos ei asetettu, käytetään oletusta).
	Maksimipituus	Määrittää tekstin suurimman sallitun pituuden (jos ei asetettu, käytetään oletusta).
	Salli BBCode	Määrittää, saako tekstissä käyttää phpBB:n muotoilukomentoja (BBCode).
	Koko teksti	Määrittää, näytetäänkö teksti kokonaan.
	Näytettävät merkit	Määrittää, kuinka monta merkkiä näytetään ennen Lue lisää -linkkiä, mikäli edellinen valinta ei ole käytössä.
	Pikavastaustyyli	Määrittää sisältöä lähetettäessä näytettävän tekstilaatikon tyylin. Pikavastaustyyliissä ei ole muotoilupainikkeita.
	Oletusarvo	Määrittää tekstilaatikon oletusarvon.
Kuva	Salli paikalliset kuvat	Määrittää, voidaanko kuvaelementissä käyttää foorumin hakemistossa olevia kuvia.
	Paikallisen kuvan kansio	Määrittää, mistä kansioista paikalliset kuvat haetaan, jos ne ovat käytössä.

	Salli linkitetyt kuvat	Määrittää, voidaanko kuvaelementissä käyttää kuvien URL-osoitteita.
	Kuvan osoitekentän pituus	Määrittää tekstilaatikon koon sisältöä lähetettäessä.
	Kuvan osoitteen maksimipituus	Määrittää suurimman sallitun määrän merkkejä linkitetyn kuvan osoitteessa.
	Kuvan leveys	Määrittää kuvan pienimmän ja suurimman sallitun leveyden pikseleissä.
	Kuvan korkeus	Määrittää kuvan pienimmän ja suurimman sallitun korkeuden pikseleissä.
	Skaalaa kuvat	Määrittää, sovitetaanko väärän kokoiset kuvat sallittuihin rajoihin vai estetäänkö niiden lähettäminen kokonaan.
	Oletuskuva	Määrittää kuvakentän oletuskuvan.
Valinta	Valintaelementin tyyppi	Määrittää valintaelementin ulkonäön. Vaihtoehtoina ovat valintanapit ja pudotusvalikko.
	Rivien määrä	Määrittää monivalintapudotusvalikon kerrallaan näkyvissä olevien rivien määrän. Jos elementti ei ole monivalintapudotusvalikko, tällä arvolla ei ole mitään merkitystä (ei pakollinen).
	Valittavissa	Määrittää, kuinka monta kohdetta käyttäjä voi enimmillään valita. Tämä vaikuttaa myös siihen, käytetäänkö valintaruutuja vai -painikkeita (jos valintaelementin tyyppi on valintanappi) tai onko se monivalintapudotusvalikko vai tavallinen (jos sen tyyppi on pudotusvalikko).

	Vaihtoehdot	Määrittää valintaelementissä valittavissa olevat vaihtoehdot.
	Oletusvalinta	Määrittää, mikä vaihtoehdoista on oletusarvoisesti valittuna.
Ryhmä	Ryhmän tyyppi	Määrittää, minkälainen ryhmä on kyseessä. Vaihtoehtoina HTML-kielen erilaiset listat, sisältöjä kokoava sisältölista ja alasiivu.
	Järjestä automaattisesti	Määrittää, järjestetäänkö ryhmän kohteet automaattisesti kun ryhmään lisätään kohde.
	Järjestystapa	Määrittää, järjestetäänkö ryhmän kohteet aika- vai aakkosjärjestyksessä, jos ne järjestetään automaattisesti.
	Järjestä käsin	Määrittää, voidaanko ryhmän kohteet järjestää manuaalisesti.
	Tee hakemisto	Tällä valinnalla ryhmästä voidaan tehdä automaattisesti hakemisto, jossa on linkit ryhmän eri osiin (esimerkiksi aakkosjärjestyksessä linkit niihin kohtiin, joissa kukin kirjain alkaa).
	Näytä otsikko	Määrittää, näytetäänkö ryhmän otsikko.
	Oletusnimi	Määrittää ryhmän oletusnimen.
Linkki	Linkin tyyppi	Määrittää, onko kyseessä linkki web-sivulle vai sähköpostilinkki.
	Osoitekentän pituus	Määrittää tekstilaatikon pituuden sisältöä lähetettäessä.
	Osoitteen maksimipituus	Määrittää, mikä on URL-osoitteen suurin sallittu pituus sisältöä lähetettäessä.
	Avaa uuteen ikkunaan	Määrittää, avataanko linkki uuteen ikkunaan (tai välilehteen) vai samaan ikkunaan.

	Oletusarvo	Määrittää linkkikentän oletusarvon.
Aika	Ajan tyyppi	Määrittää, onko kyseessä ajankohta (päivämäärä, kellonaika) vai kesto.
	Paikallinen aika	Määrittää, onko kyseessä paikallinen aika, jolloin ajankohta näkyy samana käyttäjän asetuksista riippumatta, vai sovitetaanko näytettävä aika käyttäjän aikavyöhykkeen mukaan, jolloin ajankohta muuttuu käyttäjän asetusten mukaisesti.
	Näytä paikallinen aika	Määrittää, näytetäänkö ajankohdan yhteydessä sen olevan paikallista aikaa, mikäli kyseinen toiminto on käytössä.
	Salli kesäaika	Määrittää, voiko ajankohdan lähettäjä asettaa ajankohdan olevan kesäaikaa.
	Näytä kesäaika	Määrittää, näytetäänkö ajankohdan yhteydessä sen olevan kesäaikaa, mikäli kyseinen toiminto on käytössä.
	Oletusarvo	Määrittää kentän oletusajankohdan.
Moduuli	Käytössä oleva moduuli	Moduulikentät antavat sijoittaa moduulityyppisen sisältötyypin toiseen sisältötyyppiin kentän tapaan. Tämä valinta määrittää, mitä moduulia käytetään.
	Moduulin tyyppi	Määrittää, millä tavalla moduuli valitaan.
	Sisältöjen määrä	Määrittää, kuinka monta moduulin sisältämää sisältöä moduulielementissä näytetään (ei pakollinen). Moduulityypin sisällöt voivat olla yhteisiä kaikille sisältötyypeille, joten samat sisällöt voivat näkyä useassa eri paikassa tätä käytettäessä.



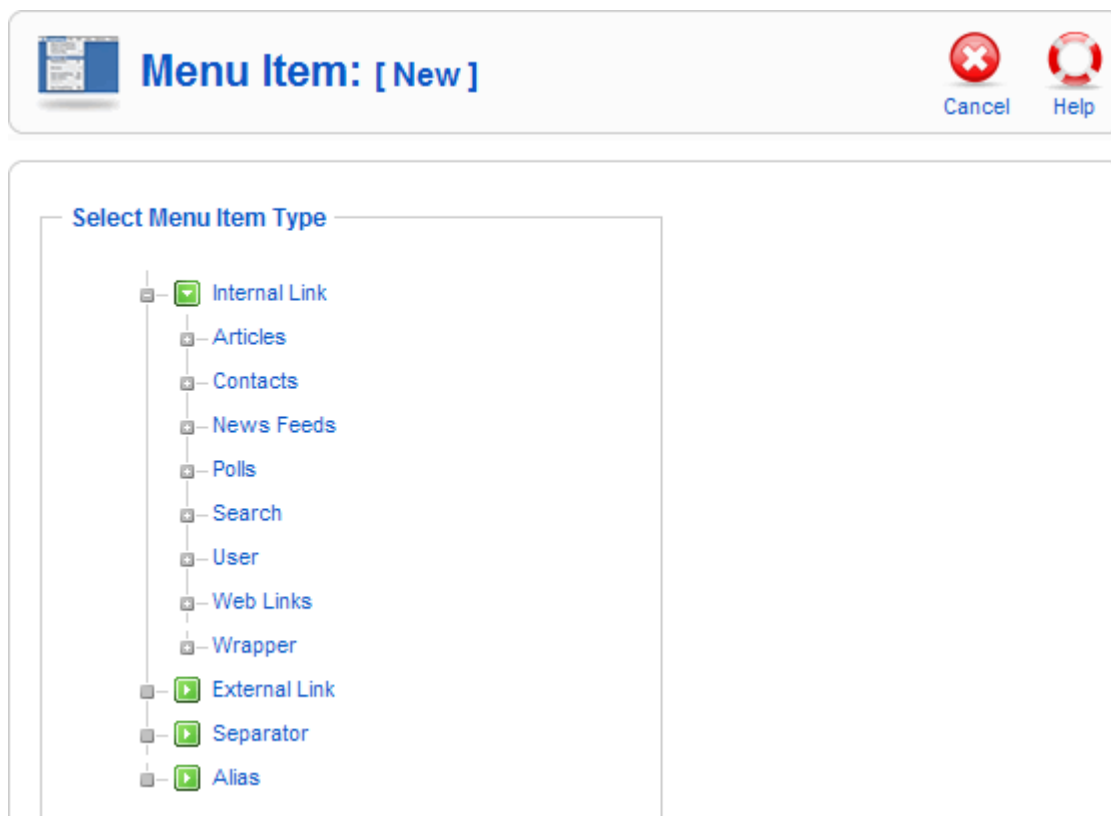
	Sisältöjen järjestys	Määrittää, missä järjestyksessä moduulielementin sisällöt näytetään.
	Oletusarvo	Määrittää oletusarvoisesti valittuna olevan moduulielementin sisällön.

#### 4.3.7 Valikkojen hallinta

Valikkojen hallinta on olennainen osa web-sisällönhallintaa, liikutaanhan verkkosivuilla juuri valikoissa olevien linkkien avulla. Valikkojen rakentaminen käsityönä on kuitenkin varsin puuduttavaa puuhaa, joten järjestelmän on syytä pystyä tekemään se automaattisesti. Tästä syystä olenkin tehnyt tietokokonaisuuksien hallintaan valinnan ”näytä valikossa”, jonka valitsemalla kyseinen tietokokonaisuus, tai oikeastaan siitä muodostettu linkki kyseiseen tietokokonaisuuteen, esitetään sivuston oletusvalikossa.

Aina oletusvalikko ei kuitenkaan vastaa tarpeita, ja toisinaan voi olla tarpeen lisätä myös valikkolinkkejä jotka eivät ole sivuston tietokokonaisuuksia, esimerkiksi linkkejä yhteistyökumppanien sivustoille (kohderyhmän ylläpitäjillä on usein ”yhteistyökumppaneita”, vaikkakin termi tarkoittaa tässä yhteydessä eri asiaa kuin liike-elämässä: yhteistyökumppani on henkilö, jonka kanssa vaihdetaan kuulumisia, ja jolta voi pyytää apua tai jota voi neuvoa tarpeen vaatiessa). Tämän vuoksi on syytä lisätä ylläpitäjälle mahdollisuus luoda myös omia valikkoja.

Valikkojen hallintaan olen ottanut mallia Joomla!-järjestelmästä. Vaikka kyseinen järjestelmä onkin omaan makuuni vaikeasti käytettävä ja aivan liian jäykkä, valikkojen luonti on toteutettu siinä esimerkillisesti. Kun ylläpitäjä lisää uuden elementin valikkoon, järjestelmä antaa listan erilaisista vaihtoehdoista, minkä tyyppinen linkki kyseessä voi olla (kuva 9). Tämänkaltainen tapa lisätä linkkejä tuntuu erittäin käyttäjäystävälliseltä, ja yhdistettynä automatisoituun oletusvalikon luontiin se tarjoonnee riittävän monipuolisen ja joustavan tavan luoda mahdollisimman monipuolisesti kaikki ylläpitäjän tarvitsemat valikkorakenteet ja niiden valikkokohteet. Tiedot valikoista ja niiden elementeistä tallennetaan tietenkin omaan tietokantatauluunsa (taulukko 13).



Kuva 9. Joomla!n valikkoelementtien luonti (lähde: Joomla.org, Screen.menus.edit.15). Uuden valikkoelementin lisääminen avasi Joomla! 1.5:ssä tällaisen näkymän, jossa sai valittua lisättävän valikkoelementin tyyppin. Sittemmin näkymä muuttui palstoitetuksi luetteloksi, jossa eri elementtityypit oli vain ryhmitelty eri otsikkojen alle, mutta syystä tai toisesta pidän itse tästä esitystavasta enemmän.

Taulukko 13. ACMS:n valikkorakenteiden tiedot. Osa tiedoista on sidoksissa siihen, mikä on valikkoelementin tyyppi, osa on yhteisiä kaikille.

Tyyppi	Ominaisuus	Selitys
Valikko	Valikon nimi	Määrittää nimen, jolla valikko näkyy.
	Valikon sijainti	Määrittää sijainnin, johon valikko ulkoasussa sijoitetaan. Sijaintien tiedot tallennetaan erikseen tietokantaan ja tyyli pohjiin.
	Valikon oikeudet	Määrittää käyttöoikeudet valikolle.
	Valikon tyyppi	Päävalikko muodostetaan automaattisesti, muut valikot ovat ylläpitäjän rakentamia.

Valikkoelementti	Elementin nimi	Määrittää valikkoelementin linkkinimen, mikäli se ei ole sisältötyyppi.
	Elementin tyyppi	Määrittää valikkoelementin tyyppin. Se voi olla linkki ulkoiselle sivustolle, jokin muu erillisesti määritelty linkki, linkki sisältötyyppiin, sisältöön, keskustelualueeseen tai viestiketjuun, tai esimerkiksi pelkkä kuva, joka vain näytetään valikkopalkissa jostain syystä.
	Elementin osoite	Määrittää osoitteen, jonne elementistä siirrytään, jos se on erillinen linkki.
	Sisäinen osoite	Määrittää kohteen, johon elementistä siirrytään, mikäli se on foorumin osa.
	Sisältö	Määrittää elementin sisällön, jos se ei olekaan linkki vaan jotakin muuta.

#### 4.3.8 Etusivun hallinta

Etusivu on se osa sivustoa, jonka kävijä näkee ensimmäiseksi saapuessaan sivustolle. Keskustelufoorumeilla tämä on useimmiten lista keskustelufoorumin keskustelualueista, näin myös phpBB-foorumeilla (kuva 10). Etusivu on käytännössä mikä tahansa sivu, joka siis vain sattuu näkymään ensimmäisenä sivustolle tultaessa. Etusivun hallinta voi siis merkitä käytännössä joko sitä, että järjestelmän tietokantaan on erikseen tallennettu sivu ”etusivu”, jolle sitten tehdään määitykset erikseen, tai sitä, että järjestelmässä on mahdollista valita jokin tietty sivu toimimaan etusivuna. Asiaa harkittuani päädyin jälkimmäiseen vaihtoehtoon, sillä tällöin etusivun vaihtaminen halutessa on helpompaa kuin silloin, jos etusivu rakennetaan erikseen olemaan erityisesti etusivu.

phpBB® Infiltrator  
creating communities Forumien modien soluttamiseen

Etsi... Tarkennettu haku

Tyhjenna

UKK Chat Rekisteröidy Kirjaudu sisään

Tänään on To Joulu 18, 2014 2:16 am

Näytä vastaamattomat viestit • Näytä aktiiviset viestiketjut

ENSIMMÄINEN KATEGORIA	Aiheet	Viestit	Uusin viesti
<b>Testialue 1</b> Tämä on testialue. Sisällytyt: D Test forum 4, D Test forum 6	68	1551	Kirjoittaja tester_12 Su Loka 21, 2012 3:49 pm
<b>Test forum 2</b>	12	352	Kirjoittaja tester_21 Ma Loka 08, 2012 8:48 pm

TEST CATEGORY 1	Aiheet	Viestit	Uusin viesti
<b>Test forum 1</b>	15	365	Kirjoittaja tester_2 To Joulu 18, 2014 2:14 am
<b>Test forum 3</b>	18	495	Kirjoittaja tester_2 To Joulu 18, 2014 2:14 am
<b>Test forum 5</b>	15	340	Kirjoittaja tester_50 Ma Loka 08, 2012 9:07 pm

KIRJAUDU SISÄÄN • REKISTERÖIDY

Käyttäjätunnus:  Salasana:  | Kirjaudu automaattisesti sisään.  **Kirjaudu sisään**

PAIKALLAOLIJAT

Yhteensä 1 käyttäjää paikalla :: Ei rekisteröityneitä, ei yhtään piilotettua ja 1 vierailija (Tieto perustuu viimeisen 5 minuutin aikana olleisiin aktiivisiin käyttäjiin)  
Eniten yhtäaikaista käyttäjiä on ollut 3 kpl, Su Loka 21, 2012 2:55 pm

Rekisteröityneet käyttäjät:  
Ei rekisteröityneitä käyttäjiä  
Selite: *Ylläpitäjät, Valvojat*

TILASTOT

Viestejä yhteensä 3103 • Viestiketjuja on yhteensä 128 kappaletta • Käyttäjiä yhteensä 51 • Uusin käyttäjä **tester\_50**

Etusivu Henkilökunta • Poista evästeet • Kaikki ajat ovat UTC + 2 tuntia

Keskustelufoorumien moottorina toimii phpBB® Forum Software © phpBB Group  
Käännös: Lurtinen, www.phpbbsuomi.com  
Time : 0.459s | 13 Queries | GZIP : Off

Kuva 10. phpBB-foorumien etusivu. Kuvassa olevalle foorumille on asennettu muutama lisäosa, jotka lisäävät joitakin asioita yläreunassa olevaan valikkoon, mutta pääsääntöisesti muokkaamattoman phpBB:n etusivu näyttää kutakuinkin tällaiselta.

Luonnollisestikin etusivun voi valitsemallanikin tavalla rakentaa erityisesti etusivuksi, mutta se voidaan tallentaa kuten mikä tahansa muukin sivu, ja kuten todettua, sitä voidaan tarvittaessa vaihtaa, jos esimerkiksi sivustolle tulijan halutaan näkevän ensimmäiseksi jokin tiedote ilman, että varsinaista etusivua siltikään menetetään. Tällöin on tietysti jossakin kohtaa järjestelmää oltava vaihtoehto, jolla voidaan valita mikä sivu esitetään sivuston etusivuna. Itse päädyin siihen, että sijoitan tämän valinnan sivuston yleisiin asetuksiin, koska siellä on muutenkin koottuna sivustoon yleisesti vaikuttavat määrittelyt. Toinen vaihtoehto olisi ollut sijoittaa valinta sivujen hallintaan, mutta yleisissä asetuksissa valinta vaikutti enemmän järkeenkäyvältä, säädetäänhän siellä nimenomaisesti sivuston yleisiä asetuksia, joihin etusivukin kaiken järjen mukaan kuuluu.

## 5 PROJEKTIN ONNISTUMISEN ARVIOINTI

Projektin valmistuttua runsaasti aikataulustaan myöhässä on todettava, että suurimmaksi ongelmaksi osoittautui juuri aikataulu. Määrittelemäni aikarajat osoittautuivat ilmeisen riittämättömiksi, sillä kesken projektin tekemisen tapahtui asioita, jotka joko vaativat välttämättä huomiotani muualle tai muuten estivät projektin etenemisen. Niinpä opin näytetyö jäi ajoittain liiankin taka-alalle. Näitä ongelmia olisi kuitenkin voinut osittain ennakoida rytmittämällä työtä eri tavalla ja tekemällä aikataulun osin joustavammaksi, jolloin aikarajojen ylitykset eivät olisi olleet samassa määrin tarpeen. Osa matkan varrella ilmenneistä seikoista oli kuitenkin siinä määrin yllättäviä, että niitä ei olisi kaikkia millään tavoin voinut välttää, koska niitä oli täysin mahdotonta edes odottaa.

Suunnittelussa tulikin vastaan monenlaisia mutkia, joita kaikkia ei vielä ole saatu selvitettyä. Työtä hidastivat lukuisat epäonniset sattumukset mukaan lukien se, että sivusto, jolle järjestelmä oli alun alkaen suunniteltu, joutui sulkemaan keskustelufooruminsa sivutilaan liittyvien ongelmien vuoksi, mikä luonnollisesti söi jonkin verran motivaatiota. Tämä ja muut vastaavat tapaukset, kuten itseni ulkopuolisista syistä syntynyt tarve muuttaa kaikki sivustoni kiireellä kahteenkin kertaan eri sivutilasta toiseen, saivat aikaan sen, että aikataulu ylittyi useaan otteeseen ja uhkasivat jopa koko työn valmistumista. Voidaan kuitenkin kysyä, että mikä se sellainen ohjelmistoprojekti on, joka ei ylitä omaa aikatauluaan?

Toinen suuri ongelma oli aikataulun pohjaksi valittu etenemistapa, joka osoittautui itselleni täysin mahdottomaksi noudattaa. Olin suunnitellut eteneväni projektien tavanomaiseen etenemistapaan vesiputousmallin mukaisesti vaiheittain siten, että yksi vaihe suoritetaan loppuun ennen seuraavan aloittamista, mutta havaitsin tämän mallin itselleni toimimattomaksi. Minun oli huomattavasti helpompi ajatella tietokantaa käyttöliittymän kautta ja toimintojenkin kulkua käyttöliittymänä kuin suunnitella ensin nämä vaiheet ja vasta sitten visuaalinen osuus. Niinpä päädyin tekemään eri vaiheita osittain rinnakkain ja päällekkäin. Tästä oli se etu, että yhdessä vaiheessa epäonnistuneeksi osoittautuva ratkaisu oli helppo korjata jokaiseen niistä vaiheista, joihin se vaikutti, eikä näin ollen ollut tarvetta palata jo valmistuneisiin vaiheisiin niiden valmistumisen jälkeen ilmenneiden ongelmien vuoksi. Haittapuolena oli asioiden osittainen sekoittuminen, ajoittai-

nen epävarmuus siitä, mikä työ kuului mihinkin vaiheeseen, ja epätietoisuus siitä, mikä vaihe on jo valmis ja mikä vielä kesken. Lopulta kävi niin, että tiesin vasta projektin päätyttyä missä järjestyksessä vaiheet olivat valmistuneet.

Suurin puute, tietysti aikataulun ylittymisen lisäksi, on käyttöliittymän testauksen jääminen puuttumaan. Tämä johtui, paitsi demoversion valmistamisessa ilmenneistä ongelmista, myös siitä, että alkuperäinen testiryhmäkseni tarkoitettu joukko hajosi, eikä uuden rekrytoiminen minkäänlaisen mielekkään aikataulun puitteissa ollut mahdollista. Harkitsin jakavani järjestelmän alpha-version phpBB:n yhteisölle, mutta kun sen valmistuminen viivästyi erinäisten pienten teknisten yksityiskohtien vuoksi päädyin siihen, että on parempi saattaa työ valmiiksi ilman testausvaihetta kuin jättää sen valmistuminen riippuvaiseksi koodauksesta, joka ei edes ole osa työtä. Näin ollen osa mallikuvista on tehty vain toimintaluonnosten pohjalta yksinomaan mallikuvien ottamista varten, eivätkä ne näin ollen ole täysin toimivia lomakkeita, eivät mahdollisesti edes täysin yhteneväisiä toistensa kanssa tekstisisältöjensä osalta, mutta koska niiden tarkoituksena onkin esitellä järjestelmän toimintaperiaatetta ja käyttöliittymää, en usko tästä olevan erityisen suurta haittaa, eihän kuvissa muutenkaan ole samaa toiminnallisuutta kuin varsinaisessa käyttöliittymässä.

Käyttöliittymä on kuitenkin suunniteltu mahdollisimman yhdenmukaiseksi phpBB:n oman käyttöliittymän kanssa, joskin lisäsin muutamia yksityiskohtia joustavuuden parantamiseksi, joten en usko sen käytettävyydeltään kovin paljoa pelkästä phpBB:stä poikkeavan, vaikka tutkimustieto asiasta jäikin puuttumaan ja tämä on näin ollen yksinomaan oma arvioni asiasta. Koska kuitenkin itse kuulun järjestelmän kohderyhmään, voisin kuvitella omallakin arviollani olevan jotain painoarvoa asiassa, olkoonkin että tietysti tunnen järjestelmän perinpohjaisesti sen itse tehtyäni, enkä näin ollen pysty näkemään siinä olevia epäloogisuuksia samalla tavoin kuin joku, joka alkaa käyttää sitä tietämättä etukäteen miten sitä oikeastaan kuuluisi käyttää.

Merkittäväksi epäonnistumiseksi projektissa lienee laskettava myös yhden vaiheen, käytettävyydestestauksen, toteutumatta jääminen. Mainittu asia tosin on laadultaan sellainen, jonka ratkaisemiseksi ainakaan minä en keksi keinoa, syynä kun oli testausryhmän värväämisen ongelma potentiaalisten testaaajien omien kiireiden vuoksi. Mahdollista olisi varmaankin ollut testaaajien etsiminen laajemmalta alueelta kuin miltä heitä etsin, mutta

se ei olisi ollut tämän projektin puitteissa mahdollista, sillä tahdoin säilyttää luottamukselliset välit minun ja testiryhmän välillä, eikä tällaista luottamusta pysty synnyttämään kovin nopealla aikataululla.

Onnistuneita seikkoja itse projektissa on vaikeampi löytää. Projekti tosin oli suunniteltu, vaikka suunnitelma ei pitänytäkään, mikä varmaankin on laskettava positiiviseksi asiaksi, ja kaikki vaiheet tuli suoritettua ainakin teoriasolla, lukuun ottamatta Käyttöliittymän viimeistely -vaihetta, joka oli niin sidoksissa Käyttöliittymän testaus -vaiheeseen, ettei ensinmainittua voitu suorittaa ilman viimeksi mainittua. Kuten kelpo projektin kuuluukin, suunnittelu eteni siten, että ongelmiin valikoitiin ratkaisuksi tarjolla olevista vaihtoehdoista paras, tai ainakin vähiten huono – tässä yhteydessä tarkoitan ilmaisulla tarkoitukseen sopivinta, en ota kantaa siihen mikä olisi absoluuttisesti ajatellen ratkaisusta paras, sikäli kuin sellaista onkaan. Lisäksi onnistuin mielestäni laatimaan aikataulu- ja muut suunnittelutaulukot helppolukuisiksi ja visuaalisesti miellyttäväiksi, mikä tosin on vain oma arvioni eikä varmastikaan projektin ja sen onnistumisen kannalta kovinkaan olennainen asia.

Onnistuminen toisaalta löytyy projektin lopputuotteesta. Lopputuloksena syntyi se mikä oli tarkoituskin, nimittäin ainakin omasta mielestäni käyttökelpoinen ja ennen kaikkea alkuehdot täyttävä suunnitelma nuorten harrastelijaylläpitäjien sisällönhallintajärjestelmäksi. Ilman tätä tosin ei välttämättä voida sanoa, että projektia olisi oikeastaan edes viety päätökseen. Vaikka suunnitelmassa onkin osia, joihin en ole tyytyväinen, nämä epäkohdat on kirjattu jo ennakkovaatimuksiin. Kehitystyötä siis tarvitaan yhä, mutta ei tämän projektin puitteissa. Ja vastapainona näille kohdille sain aikaan myös tuloksia, joista olen aidosti iloinen, näistä mainittakoon rakennepohjien rajoja asettamaton toteuttaminen.

Lopputulos on siis ainakin omasta mielestäni täysin toteuttamiskelpoinen järjestelmä, joka tosin täytyy palauttaa takaisin suunnittelupöydälle ennen kuin sitä kannattaa viedä yhtään pidemmälle. Koska phpBB 3.1 on julkaistu, ei ole järkevää laatia lisäosia phpBB 3.0:lle, jolle järjestelmän tämänhetkinen versio on suunniteltu. Tietysti olisin voinut suunnitella järjestelmän phpBB 3.1:lle, mutta olin jo aloittanut sen suunnittelun 3.0:lle, ja uuden phpBB:n version opiskelu kesken kaiken olisi vain pitkittänyt työn valmistu-

mista lisää. Näin ollen päädyin siihen, että teen ensin järjestelmän suunnittelun 3.0:lle, ja sitten kun suunnittelun tämä vaihe on valmis, voin siirtää sen 3.1:lle.

Kehitystarpeesta mainittakoon myös se, että koska tarkoituksena on tulevaisuudessa saada aikaan järjestelmä, jolla voisi hallita useampaa kuin yhtä sivustoa kerrallaan, on tähänkin kehitettävä jokin ratkaisu. Keskustelufoorumi phpBB ei sinällään tarjoa tähän mahdollisuutta, joten mitä ilmeisimmin on myös tähän tarkoitukseen kehitettävä foorumille lisäosa, jonka avulla voi yhdellä kirjautumisella kirjautua useammalle kuin yhdelle foorumiasennukselle, ellei sitten ratkaisuksi muodostu yhteen keskustelufoorumiin liitetty useampi sivusto. Tämä on kuitenkin vasta tulevaisuuden kehitystarve, eikä sen tarkempaan pohdintaan ole tässä yhteydessä tarvetta. Toinen kehitystarve ovat jotkin sivustoon liittyvät palvelut, kuten jonkinasteinen kävijöiden seuranta esimerkiksi kävijälaskurin tai sivunlatauslaskurin muodossa.

Projektiteknisesti voitaneen siis sanoa, että projekti ei ollut kovinkaan onnistunut, ellei tällainen ilmaisu ole jo karkeaa epäonnistumisen vähättelyä. Lopputuotteen kannalta tilanne on kuitenkin positiivisempi, sillä vaikka kehittämisen varaa tuotteeseen jäikin, lähtövaatimukset ja rajaus täytettiin.



## LÄHTEET

Anttila, J. 2001. Dokumenttien hallinta. Helsinki. IT Press.

Centers for Medicare & Medicaid Services (CMS). Selecting a development approach [verkkodokumentti]. 17.2.2005, tarkistettu 27.3.2008. [Viitattu 27.10.2014]. Saatavissa: <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>.

Datanamic. Introduction to database design. [verkkodokumentti]. [Viitattu 3.11.2014]. Saatavissa: <http://www.datanamic.com/support/lt-dez005-introduction-db-modeling.html>.

Hernandez, M. J. 2000. Tietokannat – suunnittelu ja toteutus. Helsinki. IT Press.

Joomla.org. Features overview. [verkkodokumentti]. Open Source Matters. [Viitattu 3.11.2014]. Saatavissa: <http://www.joomla.org/core-features.html>.

Joomla.org. Screen.menus.edit.15. [verkkodokumentti]. Open Source Matters. [Viitattu 3.11.2014]. Saatavissa: <https://docs.joomla.org/Screen.menus.edit.15>.

Joomla.org. What is Joomla? [verkkodokumentti]. Open Source Matters. [Viitattu 28.3.2009]. Saatavissa: <http://www.joomla.org/about-joomla.html>.

Kasurinen, J-P. 2013. Ohjelmistotestauksen käsikirja. Saarijärvi. Docendo.

Kohan, B. What is a Content Management System (CMS)? [verkkodokumentti]. 2010. Comentum. [Viitattu 3.11.2014]. Saatavissa: <http://www.comentum.com/what-is-cms-content-management-system.html>.

Krug, S. 2006. Älä pakota minua ajattelemaan. Helsinki. Readme.fi.

Martin, S. Effective Visual Communication for Graphical User Interfaces [verkkodokumentti]. [Viitattu 3.11.2014]. Saatavissa: [http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int\\_design.html](http://web.cs.wpi.edu/~matt/courses/cs563/talks/smartin/int_design.html).

Microsoft. Lesson 3: Optimizing the Database Design by Denormalizing [verkkodokumentti]. Microsoft. [Viitattu 3.11.2014]. Saatavissa: <https://msdn.microsoft.com/en-us/library/cc505841.aspx>.

Microsoft. Tietokannan suunnittelun perusteet [verkkodokumentti]. Microsoft. [Viitattu 27.10.2014]. Saatavissa: <https://support.office.com/fi-fi/article/Tietokannan-suunnittelun-perusteet-1eade2bf-e3a0-41b5-ae6-d2331f158280>.

Nielsen, J. 2000. WWW-suunnittelu. Helsinki. IT Press.

Page & Hughes. 1999. Oracle8. Helsinki. Suomen Atk-kustannus Oy.

- phpBB Group. About phpBB [verkkodokumentti]. phpBB Group.[Viitattu 28.3.2009]. Saatavissa: <https://www.phpbb.com/about/>.
- phpBB Group. Features of phpBB [verkkodokumentti]. phpBB Group. [Viitattu 3.11.2014]. Saatavissa: <https://www.phpbb.com/about/features/>.
- Pohjonen. R. 2002. Tietojärjestelmien kehittäminen. Jyväskylä. Docendo.
- Rihlana, S. 1997. Värioppi. Helsinki. Rakennustieto.
- Ruohonen M.J. & Salmela H. 2005. Yrityksen tietohallinto. 1.-3. painos. Helsinki. Edita Publishing Oy.
- Samela, J. 2002. Verkkosisällön hallinta. Helsinki. IT Press.
- Schade, A. Remote Usability Tests: Moderated and Unmoderated [verkkodokumentti]. 2013. Nielsen Norman Group. [Viitattu 17.12.2014]. Saatavissa: <http://www.nngroup.com/articles/remote-usability-tests/>.
- Techterms.com. CMS. [verkkodokumentti]. [Viitattu 3.11.2014]. Saatavissa: <http://techterms.com/definition/cms>.
- Techterms.com. Extranet. [verkkodokumentti]. [Viitattu 3.11.2014]. Saatavissa: <http://techterms.com/definition/extranet>.
- Techterms.com. Intranet. [verkkodokumentti]. [Viitattu 3.11.2014]. Saatavissa: <http://techterms.com/definition/intranet>.
- Veen, J. 2002. Inside Web Design. Helsinki. IT Press.
- Wiio, A. 2004. Käyttäjävälillisen sovelluksen suunnittelu. Helsinki. IT Press.

## GPL 2 -LISENSSI

Tämä on kopio phpBB:n mukana tulevasta GPL 2 -lisenssistä, ei sen julkaisijan alkuperäisestä julkaisusta. Lisenssissä kuitenkin kielletään sen muuttaminen, vaikkakin sen kopioiminen ja kopioiden julkaiseminen on kaikille sallittua, joten olettaisiin lisenssin olevan samanlainen riippumatta siitä, mistä lähteestä se onkin sitten kopioitu. Lisäksi työssä käytetään pohjana nimenomaisesti phpBB:tä, joten kyseisen järjestelmän käyttämisen lisenssin lisääminen liitteeksi vaikutti perustellulta.

GPL-lisenssistä on olemassa muitakin versioita, ja ohjelman, tai mikä tuote sitten onkin kyseessä, julkaisija voi luonnollisesti valita mitä GPL-lisenssin versiota käyttää, sillä ne poikkeavat jonkin verran toisistaan yksityiskohdiltaan, joskaan eivät yleisidealtaan.

GPL-lisenssit on suunnitellut ja julkaissut Free Software Foundation, Inc. Ne voi löytää osoitteesta <http://opensource.org/licenses/> josta ne ovat ladattavissa.

HUOMAUTUS. GPL-lisenssin mukanaolo tässä raportissa ei tarkoita, että opinnäytetyö tai edes tämä raportti olisi julkaistu kyseisellä lisenssillä (eivät ne ole); lisenssi on liitetty tähän siksi, että suunnittelun pohjana käytetyt järjestelmät käyttävät sitä, ja että lukija, joka ei tunne lisenssin sisältöä, voisi paremmin ymmärtää mistä raportin tekstiosassa puhutaan.

## GNU GENERAL PUBLIC LICENSE

Version 2, June 1991

Copyright (C) 1989, 1991 Free Software Foundation, Inc.

675 Mass Ave, Cambridge, MA 02139, USA

Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

## Preamble

The licenses for most software are designed to take away your freedom to share and change it. By contrast, the GNU General Public License is intended to guarantee your freedom to share and change free software--to make sure the software is free for all its users. This General Public License applies to most of the Free Software Foundation's software and to any other program whose authors commit to using it. (Some other Free Software Foundation software is covered by the GNU Library General Public License instead.) You can apply it to your programs, too.

When we speak of free software, we are referring to freedom, not price. Our General Public Licenses are designed to make sure that you have the freedom to distribute copies of free software (and charge for this service if you wish), that you receive source code or can get it if you want it, that you can change the software or use pieces of it in new free programs; and that you know you can do these things.

To protect your rights, we need to make restrictions that forbid anyone to deny you these rights or to ask you to surrender the rights. These restrictions translate to certain responsibilities for you if you distribute copies of the software, or if you modify it.

For example, if you distribute copies of such a program, whether gratis or for a fee, you must give the recipients all the rights that you have. You must make sure that they, too, receive or can get the source code. And you must show them these terms so they know their rights.

We protect your rights with two steps: (1) copyright the software, and (2) offer you this license which gives you legal permission to copy, distribute and/or modify the software.

Also, for each author's protection and ours, we want to make certain that everyone understands that there is no warranty for this free software. If the software is modified by someone else and passed on, we want its recipients to know that what they have is not the original, so that any problems introduced by others will not reflect on the original authors' reputations.

Finally, any free program is threatened constantly by software patents. We wish to avoid the danger that redistributors of a free program will individually obtain patent licenses, in effect making the program proprietary. To prevent this, we have made it clear that any patent must be licensed for everyone's free use or not licensed at all.

The precise terms and conditions for copying, distribution and modification follow.

## GNU GENERAL PUBLIC LICENSE

### TERMS AND CONDITIONS FOR COPYING, DISTRIBUTION AND MODIFICATION

0. This License applies to any program or other work which contains a notice placed by the copyright holder saying it may be distributed under the terms of this General Public License. The "Program", below, refers to any such program or work, and a "work based on the Program" means either the Program or any derivative work under copyright law: that is to say, a work containing the Program or a portion of it, either verbatim or with modifications and/or translated into another language. (Hereinafter, translation is included without limitation in the term "modification".) Each licensee is addressed as "you".

Activities other than copying, distribution and modification are not covered by this License; they are outside its scope. The act of running the Program is not restricted, and the output from the Program is covered only if its contents constitute a work based on the

Program (independent of having been made by running the Program).  
Whether that is true depends on what the Program does.

1. You may copy and distribute verbatim copies of the Program's source code as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the Program a copy of this License along with the Program.

You may charge a fee for the physical act of transferring a copy, and you may at your option offer warranty protection in exchange for a fee.

2. You may modify your copy or copies of the Program or any portion of it, thus forming a work based on the Program, and copy and distribute such modifications or work under the terms of Section 1 above, provided that you also meet all of these conditions:

a) You must cause the modified files to carry prominent notices stating that you changed the files and the date of any change.

b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the Program or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License.

c) If the modified program normally reads commands interactively when run, you must cause it, when started running for such interactive use in the most ordinary way, to print or display an announcement including an appropriate copyright notice and a notice that there is no warranty (or else, saying that you provide a warranty) and that users may redistribute the program under these conditions, and telling the user how to view a copy of this

License. (Exception: if the Program itself is interactive but does not normally print such an announcement, your work based on the Program is not required to print an announcement.)

These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the Program, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the Program, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it.

Thus, it is not the intent of this section to claim rights or contest your rights to work written entirely by you; rather, the intent is to exercise the right to control the distribution of derivative or collective works based on the Program.

In addition, mere aggregation of another work not based on the Program with the Program (or with a work based on the Program) on a volume of a storage or distribution medium does not bring the other work under the scope of this License.

3. You may copy and distribute the Program (or a work based on it, under Section 2) in object code or executable form under the terms of Sections 1 and 2 above provided that you also do one of the following:

- a) Accompany it with the complete corresponding machine-readable source code, which must be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,
- b) Accompany it with a written offer, valid for at least three years, to give any third party, for a charge no more than your

cost of physically performing source distribution, a complete machine-readable copy of the corresponding source code, to be distributed under the terms of Sections 1 and 2 above on a medium customarily used for software interchange; or,

c) Accompany it with the information you received as to the offer to distribute corresponding source code. (This alternative is allowed only for noncommercial distribution and only if you received the program in object code or executable form with such an offer, in accord with Subsection b above.)

The source code for a work means the preferred form of the work for making modifications to it. For an executable work, complete source code means all the source code for all modules it contains, plus any associated interface definition files, plus the scripts used to control compilation and installation of the executable. However, as a special exception, the source code distributed need not include anything that is normally distributed (in either source or binary form) with the major components (compiler, kernel, and so on) of the operating system on which the executable runs, unless that component itself accompanies the executable.

If distribution of executable or object code is made by offering access to copy from a designated place, then offering equivalent access to copy the source code from the same place counts as distribution of the source code, even though third parties are not compelled to copy the source along with the object code.

4. You may not copy, modify, sublicense, or distribute the Program except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense or distribute the Program is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such



parties remain in full compliance.

5. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to modify or distribute the Program or its derivative works. These actions are prohibited by law if you do not accept this License. Therefore, by modifying or distributing the Program (or any work based on the Program), you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or modifying the Program or works based on it.

6. Each time you redistribute the Program (or any work based on the Program), the recipient automatically receives a license from the original licensor to copy, distribute or modify the Program subject to these terms and conditions. You may not impose any further restrictions on the recipients' exercise of the rights granted herein. You are not responsible for enforcing compliance by third parties to this License.

7. If, as a consequence of a court judgment or allegation of patent infringement or for any other reason (not limited to patent issues), conditions are imposed on you (whether by court order, agreement or otherwise) that contradict the conditions of this License, they do not excuse you from the conditions of this License. If you cannot distribute so as to satisfy simultaneously your obligations under this License and any other pertinent obligations, then as a consequence you may not distribute the Program at all. For example, if a patent license would not permit royalty-free redistribution of the Program by all those who receive copies directly or indirectly through you, then the only way you could satisfy both it and this License would be to refrain entirely from distribution of the Program.

If any portion of this section is held invalid or unenforceable under any particular circumstance, the balance of the section is intended to

apply and the section as a whole is intended to apply in other circumstances.

It is not the purpose of this section to induce you to infringe any patents or other property right claims or to contest validity of any such claims; this section has the sole purpose of protecting the integrity of the free software distribution system, which is implemented by public license practices. Many people have made generous contributions to the wide range of software distributed through that system in reliance on consistent application of that system; it is up to the author/donor to decide if he or she is willing to distribute software through any other system and a licensee cannot impose that choice.

This section is intended to make thoroughly clear what is believed to be a consequence of the rest of this License.

8. If the distribution and/or use of the Program is restricted in certain countries either by patents or by copyrighted interfaces, the original copyright holder who places the Program under this License may add an explicit geographical distribution limitation excluding those countries, so that distribution is permitted only in or among countries not thus excluded. In such case, this License incorporates the limitation as if written in the body of this License.

9. The Free Software Foundation may publish revised and/or new versions of the General Public License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns.

Each version is given a distinguishing version number. If the Program specifies a version number of this License which applies to it and "any later version", you have the option of following the terms and conditions either of that version or of any later version published by the Free

Software Foundation. If the Program does not specify a version number of this License, you may choose any version ever published by the Free Software Foundation.

10. If you wish to incorporate parts of the Program into other free programs whose distribution conditions are different, write to the author to ask for permission. For software which is copyrighted by the Free Software Foundation, write to the Free Software Foundation; we sometimes make exceptions for this. Our decision will be guided by the two goals of preserving the free status of all derivatives of our free software and of promoting the sharing and reuse of software generally.

#### NO WARRANTY

11. BECAUSE THE PROGRAM IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE PROGRAM, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE PROGRAM "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAM IS WITH YOU. SHOULD THE PROGRAM PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

12. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING

WILL ANY COPYRIGHT HOLDER, OR ANY OTHER PARTY WHO MAY  
MODIFY AND/OR  
REDISTRIBUTE THE PROGRAM AS PERMITTED ABOVE, BE LIABLE TO  
YOU FOR DAMAGES,  
INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL  
DAMAGES ARISING  
OUT OF THE USE OR INABILITY TO USE THE PROGRAM (INCLUDING  
BUT NOT LIMITED  
TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR  
LOSSES SUSTAINED BY  
YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAM TO OPERATE  
WITH ANY OTHER  
PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN  
ADVISED OF THE  
POSSIBILITY OF SUCH DAMAGES.

END OF TERMS AND CONDITIONS

## JÄRJESTELMÄN TEHTÄVÄT JA TOIMINNOT

### Järjestelmän tarkoitus

ACMS-sisällönhallintajärjestelmän tarkoituksena on toimia välineenä web-sivuston sisällön julkaisemiseen ja sivuston rakentamiseen siten, että ylläpitäjän ei tarvitse huolehtia itse fyysisten sivujen luomisesta, vaan järjestelmä tuottaa ne tietokantaan tallennetun tiedon perusteella yhdistettynä tyyliohjeiden esitysohjeisiin.

### Järjestelmän tehtävät

#### 1. Tietokokonaisuuksien hallinta

**Toiminto:** Kategorioiden ja sivujen luonti

**Toiminnon tarkoitus:** luoda ryhmittelyelementtejä, joiden avulla sisältöelementit luokitellaan, ja luoda sivuja, joiden avulla sisältö esitetään.

**Toiminnon kuvaus:** Hallintapaneelissa luodaan uusi kategoria, jos halutaan luoda erillinen ryhmittelykokonaisuus johon voidaan sisällyttää sivuja, tai sivu, mikäli tahdotaan luoda sivu.

- Kategorian ainoa tarkoitus on toimia ryhmittelyelementtinä valikoissa ja muissa vastaavissa paikoissa.
- Sivulle voidaan sijoittaa muuta sisältöä.

**Toiminto:** Kategorioiden ja sivujen hallinta

**Toiminnon tarkoitus:** muokata kategorioiden ja sivujen järjestystä.

**Toiminnon kuvaus:** Hallintapaneelissa siirretään kategoriat ja sivut eri järjestykseen ylläpitäjän haluamalla tavalla. Niiden järjestystä voidaan muuttaa saman tason sisällä, tai siirtää ne eri tasoille, tai toisen kategorian alaisiksi.

**Toiminto:** Kategorioiden muokkaus

**Toiminnon tarkoitus:** käsitellä luokkia, joihin sisällöt luokitellaan.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan kategoria, jota tahdotaan muokata. Tämän asetukset muutetaan halutunlaisiksi.

- Kategorialla itsellään ei ole paljoakaan asetuksia.
- Kategoria kuitenkin sisältää sivuja ja mahdollisesti muita kategorioita. Näitä tulee päästä muokkaamaan kategorian kautta.

**Toiminto:** Sivujen muokkaus

**Toiminnon tarkoitus:** käsitellä sivuja, joiden kautta sisältö esitetään.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan sivu, jota tahdotaan muokata. Tämän asetukset muutetaan halutunlaisiksi.

- Sivun nimi ja kuvaus esitetään sivustolla sivun otsikkona ja otsikkokappaleena.
- Sivulle voi lisätä muita sisältötyyppejä, ja näin tuleekin tehdä mikäli haluaa sillä olevan oikeasti jotakin sisältöä.

**Toiminto:** Kategorioiden ja sivujen poistaminen

**Toiminnon tarkoitus:** jos sivustolla on turhia kategorioita tai sivuja, ne voidaan poistaa tämän toiminnon avulla, jotta rakenne pysyy selkeänä.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan sivu tai kategoria, joka tahdotaan poistaa. Järjestelmä varmistaa poiston, ennen kuin mitään lopullista tehdään, siltä varalta että sivu tai kategoria on yritetty poistaa vahingossa.

## 2. Tekstisisällön hallinta

**Toiminto:** Tekstien luonti

**Toiminnon tarkoitus:** lisätä sivustolle tekstejä katseltavaksi.

**Toiminnon kuvaus:** Hallintapaneelissa siirrytään tekstisisältöjen hallintaan. Teksti kirjoitetaan, sille asetetaan tarpeelliset tiedot ja se tallennetaan.

**Toiminto:** Tekstien muokkaus

**Toiminnon tarkoitus:** jos teksti ei enää pidä paikkaansa, tai jos siinä on virheitä, sitä voidaan muokata olemaan jälleen sellainen kuin sen pitäisi olla.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan muokattava teksti. Muokkaukset tehdään, ja lopputulos tallennetaan.

**Toiminto:** Tekstien poistaminen

**Toiminnon tarkoitus:** jos tekstiä ei ole enää tarpeen esittää sivulla, se voidaan poistaa sieltä häiritsemästä ja olemasta tiellä.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan poistettava teksti. Järjestelmä kysyy varmistuksen ennen tekstin poistamista, jotta vahinkopoistoilta vältytään.

**Toiminto:** Tekstien lisääminen kategorioihin ja sivuihin

**Toiminnon tarkoitus:** jos tekstiä ei ole lisätty mihinkään ryhmään, josta sivut rakennetaan, ei sitä myöskään voida näyttää sivustolla. Lisäämällä teksti johonkin kategoriaan tai sivuun se myös julkaistaan.

**Toiminnon kuvaus:** Hallintapaneelista valitaan haluttu teksti. Sille valitaan sivu, jolla se halutaan näyttää. Muutokset tallennetaan, ja teksti ilmaantuu sivustolle.

### 3. Uutisten hallinta

**Toiminto:** Uutisten luonti

**Toiminnon tarkoitus:** uutisilla ylläpitäjä voi tiedottaa kävijöilleen sivuston uusista tapahtumista, esimerkiksi päivitystiedotteista, tai antaa muita tiedotteita.

**Toiminnon kuvaus:** Hallintapaneelista uutisten hallintasivulta valitaan uutisen lähettäminen. Haluttu teksti kirjoitetaan, ja uutisen ajankohta kirjautuu automaattisesti sen lähetyksajankohdaksi, kun uutinen tallennetaan.

**Toiminto:** Uutisten muokkaus

**Toiminnon tarkoitus:** jos uutisessa on kirjoitusvirhe, jos asia ei ole enää ajankohtainen (mutta uutista ei tahdota poistaa) tai jos asiaa tulee lisää, mutta ei niin paljoa että kannattaisi lähettää uusi uutinen, uutista voidaan muokata tietojen päivittämiseksi.

**Toiminnon kuvaus:** Ylläpitäjä valitsee päivitettävän uutisen, ja ottaa käyttöön muokkaustoiminnon. Halutut kohdat muutetaan. Halutessaan ylläpitäjä voi valita, että uutisen yhteydessä näytetään sen muokkausajankohta.

**Toiminto:** Uutisten poistaminen

**Toiminnon tarkoitus:** jos uutinen on vanhentunut, eikä sitä haluta jättää sivustolle roikkumaan ja hämmentämään kävijöitä, se voidaan poistaa.

**Toiminnon kuvaus:** Ylläpitäjä valitsee poistettavan uutisen ja ottaa käyttöön poistotoiminnon. Järjestelmä pyytää varmistusta ennen kuin lopullinen poisto tehdään, jotta vahingossa suoritetuilta poistoilta vältytään.

**Toiminto:** Uutisten liittäminen sivuihin

**Toiminnon tarkoitus:** jotta uutisia voidaan lukea, ne on myös esitettävä.

**Toiminnon kuvaus:** Hallintapaneelissa lisättäessä sisältöä sivuille voidaan valita, että jossakin kohtaa näytetään uutisia. Tähän voidaan määrittää näytettäväksi uusimmat uutiset tai jokin tietty uutinen.

#### 4. Valikkojen hallinta

**Toiminto:** Valikkoelementtien määrittäminen

**Toiminnon tarkoitus:** jos ylläpitäjä tahtoo luoda omia valikkoelementtejään, ne on määritettävä erikseen.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan valikon hallintaosio. Siellä lisätään valikkoelementti haluttuun valikkoon. Tämä tallennetaan, ja sivuston valikko päivittyy.

**Toiminto:** Valikkoelementtien muokkaus

**Toiminnon tarkoitus:** jos valikkoelementin kohteen nimi tai osoite muuttuvat, on syytä päivittää myös valikkoelementti, jotta se johtaa oikeaan paikkaan.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan haluttu valikkoelementti. Tiedot muutetaan ja tallennetaan, ja päivitys astuu voimaan.

**Toiminto:** Valikkoelementtien poistaminen

**Toiminnon tarkoitus:** jos valikkoelementin kohde ei ole enää olemassa, tai jos sitä ei muusta syystä haluta enää esitellä valikossa, se voidaan poistaa häiritsemästä.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan poistettava valikkoelementti ja valitaan sen poisto. Järjestelmä kysyy varmistusta ennen lopullista poistoa, jotta vahingossa suoritetuilta sivuston rampauttamisilta vältytään.



#### 4. Kuvien hallinta

**Toiminto:** Kuvien luonti

**Toiminnon tarkoitus:** jotta sivustolla voidaan esitellä kuvia, täytyy kuvat jollain tavalla määrittää ennen kuin ne voidaan esitellä yleisölle.

**Toiminnon kuvaus:** Hallintapaneelissa kuvien hallinnassa valitaan uuden kuvan lisääminen. Määritetään kuvan osoite ja tiedot sekä galleria, johon se kuuluu. Tiedot tallennetaan, ja kuva ilmaantuu sivustolle.

**Toiminto:** Kuvien muokkaus

**Toiminnon tarkoitus:** jos kuvan tiedoissa on virhe, tai niitä vain muuten halutaan korjata, kuvia on muokattava. Tämä ei tarkoita kuvan ulkonäön muokkausta vaan esimerkiksi sen nimen tai kuvauksen korjailua.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan muokattava kuva ja korjataan tiedot halutunlaisiksi. Kun korjaus on suoritettu, muutokset tallennetaan.

**Toiminto:** Kuvien poistaminen

**Toiminnon tarkoitus:** jos kuvaa ei haluta enää esitellä, se voidaan poistaa.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan poistettava kuva ja suoritetaan poisto. Järjestelmä kysyy varmistusta ennen lopullista poistoa, jottei peruuttamatonta vahinkoa sivustolle pääse tapahtumaan.

**Toiminto:** Kuvien lisääminen kuvagallerioihin

**Toiminnon tarkoitus:** kuvia voidaan ryhmitellä aihepiireittäin ja jakaa erillisille sivuille kuvagallerioiden avulla. Tämä helpottaa niiden hallintaa.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan joko haluttu galleria, ja siihen valitaan liitettävät kuvat, tai haluttu kuva, ja sille asetetaan haluttu galleria. Tiedot tallennetaan.

## 5. Sivuston asetusten hallinta

**Toiminto:** Tyylien hallinta: lisäys, muokkaus, poistaminen

**Toiminnon tarkoitus:** sivuston ulkoasua voi muuttaa muutamalla napin painalluksella. Ulkoasua ei tarvitse julkaista samalla hetkellä kun se on ladattu, vaan ylläpitäjä voi testata sitä ensin itse niin että näkee muutokset vain kirjautuneena.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan haluttu ulkoasu joka asennetaan, päivitetään tai poistetaan. Ennen poistoa järjestelmä kysyy varmistusta.

**Toiminto:** Etusivun hallinta: etusivun elementtien muokkaaminen

**Toiminnon tarkoitus:** etusivun saaminen sen mukaiseksi kuin ylläpitäjä haluaa.

**Toiminnon kuvaus:** Hallintapaneelissa valitaan elementit, jotka etusivulla halutaan näyttää. Tarvittaessa muokataan tekstit halutunlaisiksi ja tallennetaan.

**Toiminto:** Omien tietokenttien hallinta: tietokenttien luonti, muokkaus ja poistaminen, liittäminen tietokokonaisuuksiin ja sivuihin

**Toiminnon tarkoitus:** rakennepohjien saaminen ylläpitäjän haluamaan muotoon onnistuu vain, jos niiden tiedot saadaan ylläpidon haluamaan muotoon.

**Toiminnon kuvaus:** Hallintapaneelissa luodaan halutunlainen tietokenttä tai muokataan sitä. Valitaan haluttu sivu, ja lisätään tietokenttä sinne. Tarvittaessa tietokenttä poistetaan. Järjestelmä kysyy varmistusta ennen poistoa.

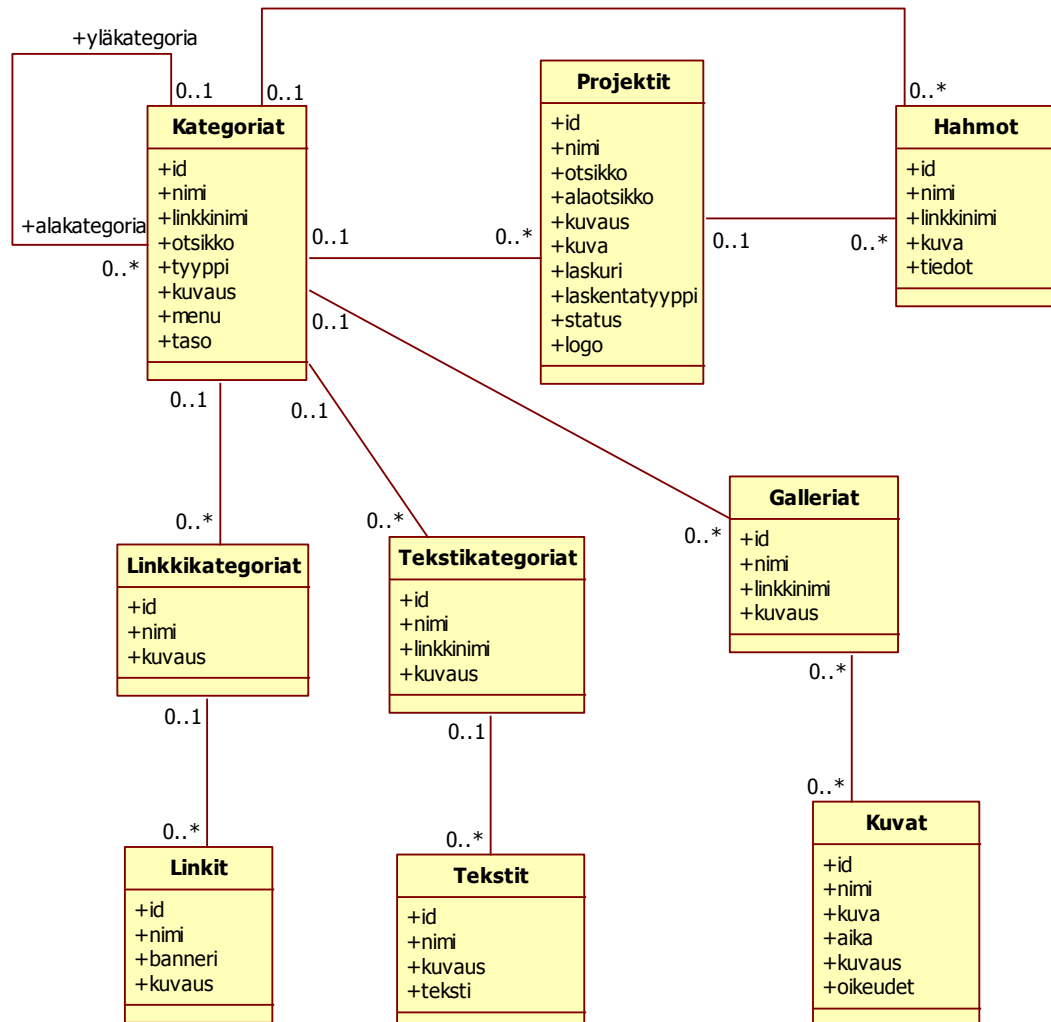
## 6. Lokitiedot

**Toiminto:** Lokitietojen keruu ja esittäminen

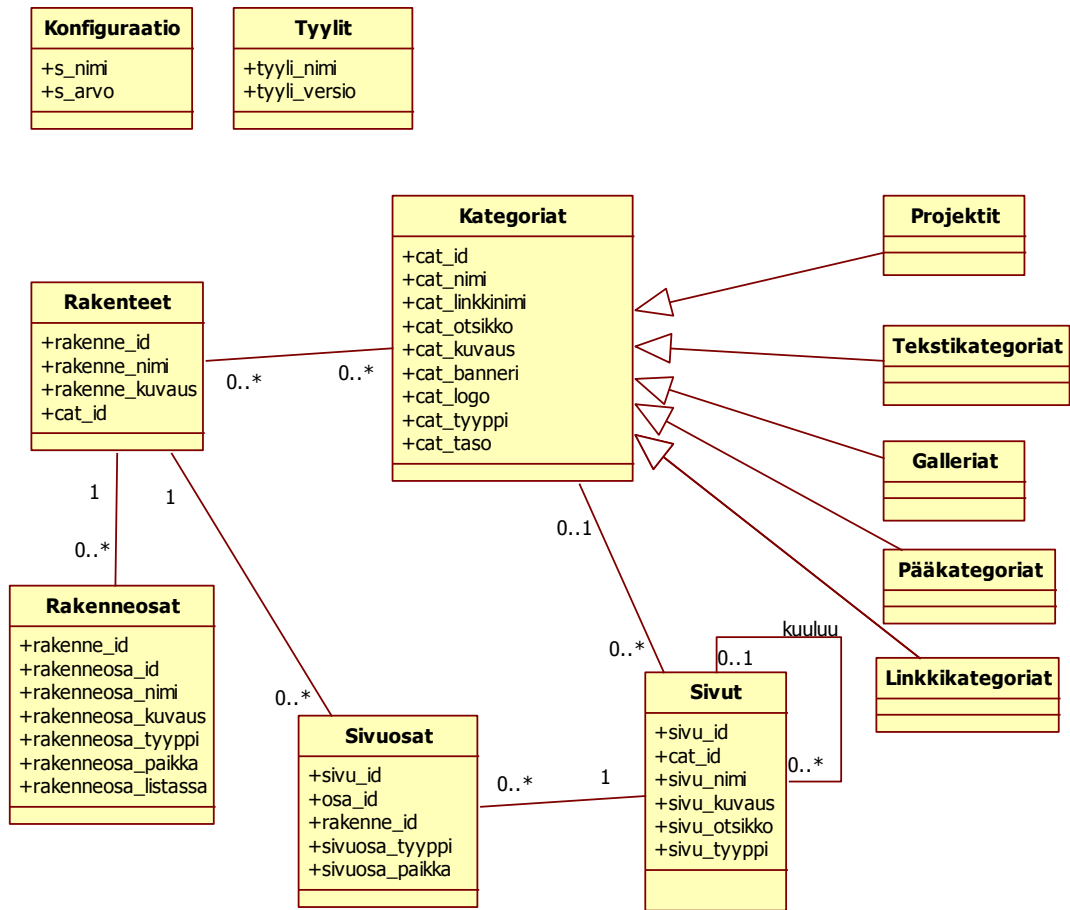
**Toiminnon tarkoitus:** ylläpitäjä voi halutessaan pitää kirjaa sivuston kävijämääristä ja sivustolla olevasta materiaalista.

**Toiminnon kuvaus:** Kävijän vieraillessa sivustolle tiedot tallentuvat kävijälaskuriin, jota voi tarkastella erillisellä laskurisivulla. Sivuston materiaalitylöt ovat nähtävissä hallintapaneelissa osastoilla, joihin kyseiset materiaalit liittyvät.

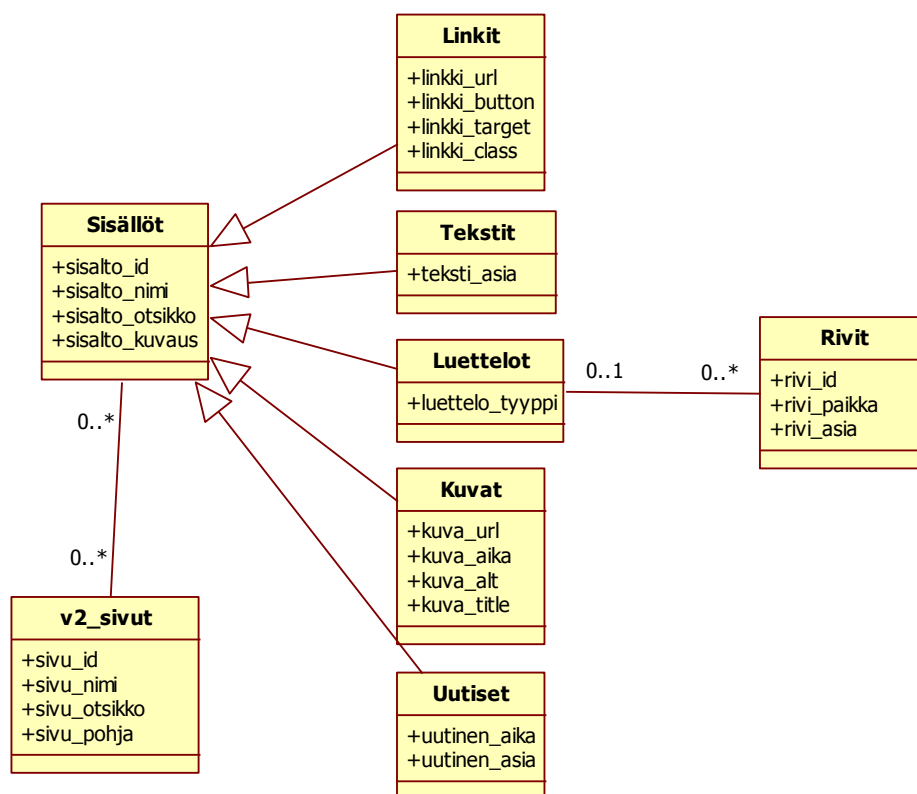
## ENSIMMÄINEN TIETOKANTAVERSIO



TOINEN TIETOKANTAVERSIO



## KOLMAS TIETOKANTAVERSIO



## ENSIMMÄISEN JÄRJESTELMÄN KÄYTTÖLIITTYMÄ

**ACMS :: ylläpitoliittymä**

### Kirjaudu sisään

Käyttäjätunnus:

Salasana:

[sivustolle](#)

ACMS is copyright © by Liisa Peippo 2009  
this site is for testing use only

**ACMS :: ylläpitoliittymä**

[Kirjaudu ulos](#) | [Ylläpitotyökalut](#) | [Kategoriat](#) | [Linkit](#) | [Kuvat](#) | [Sisällöt](#)

### Ylläpitotyökalut

#### Sivuston hallinta

[ACMS - sivuston asetukset](#) | [rakennepohjat](#) | [tyylit](#)  
[ACMS- testisivusto.](#)  
 Alakategorioita: 0 | Tekstejä: 0 | Hahmoja: 2 | Kuvia: 0

ACMS is copyright © by Liisa Peippo 2009  
this site is for testing use only

**ACMS :: ylläpitoliittymä**

[Kirjaudu ulos](#) | [Etusivu](#) | [Asetukset](#) | [Rakenteet](#) | [Kategoriat](#) | [Linkit](#) | [Kuvat](#) | [Sisällöt](#) | [Tyylit](#)

### Sivuston kategoriat

#### Pääkategoriat

**Kategorian 1 nimi** - [avaa](#) | [muokkaa](#) | [poista](#) | [siirrä ylöspäin](#) | [siirrä alaspäin](#)  
 Kategorian 1 kuvaus tulee tähän.  
 Alakategorioita: 2 | Tekstejä: 5 | Hahmoja: 2 | Kuvia: 3

**Kategorian 2 nimi** - [avaa](#) | [muokkaa](#) | [poista](#) | [siirrä ylöspäin](#) | [siirrä alaspäin](#)  
 Kategorian 2 kuvaus tulee tähän.  
 Alakategorioita: 0 | Tekstejä: 0 | Hahmoja: 2 | Kuvia: 0


**Kategorian 3 nimi** - [avaa](#) | [muokkaa](#) | [poista](#) | [siirrä ylöspäin](#) | [siirrä alaspäin](#)  
 Kategorian 3 kuvaus tulee tähän.  
 Alakategorioita: 0 | Tekstejä: 0 | Hahmoja: 2 | Kuvia: 0

Lisää uusi:

ACMS is copyright © by Liisa Peippo 2009  
this site is for testing use only

## Sivuston asetukset

## Ulkonäköasetukset

Sivuston nimi:	<input type="text" value="ACMS"/>	
Sivuston kuvaus:	<input type="text" value="testaussivusto"/>	Lyhyt kuvaus sivustosta, näytetään nimen yhteydessä.
Sivuston ulkoasu:	<input type="text" value="Wolves of the Northern Sea"/>	
Sivuston logo:		<input type="text" value="header.jpg"/>

## Kansioasetukset

Sivuston osoite:	<input type="text" value="http://127.0.0.1/wa/"/>	Sivuston osoite, juurikansio. Lopussa kauttaviiva /
Kuvat:	<input type="text" value="img/"/>	Sivuston kuvat sisältävä kansio, suhteessa juurikansioon. Lopussa kauttaviiva /
Logot:	<input type="text" value="logot/"/>	Sivuston logot sisältävä kansio, suhteessa kuvakansioon. Lopussa kauttaviiva /
Linkitettyjen sivujen buttonit:	<input type="text" value="but/"/>	Sivustolle linkitettyjen sivujen buttonit sisältävä kansio, suhteessa kuvakansioon. Lopussa kauttaviiva /
Linkitysbuttonit:	<input type="text" value="lbut/"/>	Sivuston linkitysbuttonit sisältävä kansio, suhteessa kuvakansioon. Lopussa kauttaviiva /

## Ylläpitoasetukset

Avauspäivä:	<input type="text" value="12.3.2009"/>	Päivä, jona sivusto on ensimmäisen kerran julkaistu yleisölle.
Ylläpitäjä:	<input type="text" value="Liisa Peippo"/>	Nimimerkki, jolla ylläpitäjä esiintyy.
Sähköposti:	<input type="text" value="osoite@domain.com"/>	Sähköpostiosoite, johon voi lähettää sivustoon liittyvää postia.

## Kirjautumisasetukset

**Koske näihin tietoihin vain, jos tahdot muuttaa käyttäjätunnustasi tai salasanaasi.**

Käyttäjätunnus:	<input type="text" value="admin"/>	Käyttäjätunnus, jolla kirjaudutaan ylläpitoliittymään.
Salasana:	<input type="text"/>	Kirjautumisasetusten muutokset on varmistettava salasanalla.
Uusi salasana:	<input type="text"/>	Jos tahdot vaihtaa salasanasasi, kirjoita uusi salasana tähän.
Uusi salasana uudelleen:	<input type="text"/>	Jos tahdot vaihtaa salasanasasi, kirjoita uusi salasana tähän toistamiseen.

## Linkit

### Linkkikategoriat

**Ylläpitäjän sivut** - avaa | muokkaa | poista | siirrä ylöspäin | siirrä alaspäin

Ylläpitäjän muille sivustoille vieviä linkkejä.

Linkejä: 8

**Taideyhteisöjä** - avaa | muokkaa | poista | siirrä ylöspäin | siirrä alaspäin

Ulkomaisia ja kotimaisia taideyhteisöjä.

Linkejä: 3

Lisää uusi:

Lisää

### Kategoriattomat linkit

**Kategoriattoman linkin nimi** - muokkaa | poista

Linkki, jota ei ole vielä lisätty mihinkään kategoriaan.

Lisää uusi:

Lisää



## KÄYTETTÄVYYSTESTAUKSEN SUUNNITELMA

### 1. TESTAUKSEN TOTEUTUSTAPA

Yhteydenpitoi kohderyhmään tapahtuu Internetin välityksellä keskustelufoorumi- ja sähköpostiviestein, joten testaavan henkilön käyttäytymistä ei ole mahdollista havainnoida olemalla fyysisesti samassa tilassa, vaikka testaajan havainnoimisesta olisikin hyötyä järjestelmän käytettävyyden tulkinnan kannalta (Krug 2006, 135). Tässä yhteydessä ei siis voida varsinaisesti puhua sen enempää laboratorio- kuin kenttätestauksesta, sillä kumpikin näistä malleista vaatii testattavan havainnointia. Jonkinlainen yhteys testattavaan olisi mahdollista rakentaa web-kameran ja pikaviestinohjelman kautta, jolloin testattavaa voisi tarkkailla kameran välityksellä ja pikaviestimen avulla kysellä tarkentavia kysymyksiä, mutta omat tietoliikenneyhteyteni eivät mahdollistaneet tätä. Ainoaksi testausmenetelmäksi jäi siis valvoma-ton etätestaus.

Tietokoneohjelmia testataan jatkuvasti niin, että puolivalmis ohjelma jaetaan halukaille testikäyttäjille arkikäyttöön koeteltavaksi. Vaikka tällä menetelmällä testataan-kin yleensä kokonaisia ohjelmia pelkkien käyttöliittymien sijaan, päädyin käyttämään mallia pohjana myös omassa testauksessani, sillä se on ainoa löytämäni ratkaisu joka ei edellytä välitöntä kontaktia testaajan ja testattavan välillä. Koska testauksessa on tarkoitus tutkia käyttöliittymän toimivuutta, ei etsiä virheitä ohjelmakoodista, ei testattavaksi tarjota puolivalmista järjestelmää, vaan osittain käytettävissä oleva käyttöliittymämalli, jolla testiryhmä yrittää suorittaa joitakin ylläpidollisia tehtäviä.

Testaajien on pystyttävä käyttämään käyttöliittymämallia ilman palvelinohjelmistojen asentamista omalle tietokoneelleen, ja koska mielestäni on parempi että testausta voidaan suorittaa myös ilman Internet-yhteyttä, malli on rakennettu siten, että se ei käytä tietokantaa. Tämä tosin aiheuttaa sen, että tilannetta ei voi tallentaa, eli yhden asian testaus on suoritettava yhdellä istunnolla, mutta asia tuskin aiheuttaa suurta ongelmaa. Malli jaetaan testiryhmälle pakattuna kansiona, joka sisältää paitsi

Internet-selaimella käytettävän mallin myös testauksen ohjeet. Jakamisen yhteydessä annetaan lyhyt opastus, jossa kerrotaan mistä on kysymys, mutta varsinaiset ohjeet ja testustehtävät annetaan jaettavassa paketissa. Testaajat suorittavat annetut tehtävät ja palauttavat raportin, jossa kertovat onnistumisensa ja epäonnistumisensa sekä havaitsemansa hyvät ja huonot puolet käyttöliittymässä.

## 2. TESTAUKSEN OHJEISTUS JA TESTITEHTÄVÄT

ACMS on suunnitteluvaiheessa oleva sisällönhallintajärjestelmä. Tämän testauksen tarkoituksena on tutkia kuinka helppo sitä on käyttää. Ole ystävällinen ja yritä kirjoittaa tehtäviä suorittaessasi muistiinpanoja siitä, mitä teet järjestelmän kanssa ja millaisia ajatuksia se herättää yrittäessäsi käyttää sitä. Kun olet suorittanut testauksen, lähetä nämä muistiinpanot allekirjoittaneelle, jotta voin hyödyntää testin tuloksia saadakseni järjestelmästä paremman. Testin tarkoituksena ei ole arvostella sinua tai sivustoasi, vaan saada ACMS:n käyttöliittymä paremmaksi niin, että muutkin kuin minä osaisivat käyttää sitä.

Huomautus: suunnittelen järjestelmää opinnäytetyönä. Saatan käyttää kommentteja opinnäytetyön raportissa. En kuitenkaan millään tavalla ilmaise kuka on sanonut ja mitä. Ellet halua kommenttejasi käytettävän, ilmoitathan siitä palauttaessasi raportisi.

Seuraavaksi luettelen testustehtävät. Suorita ne annetussa järjestyksessä annettujen ohjeiden mukaan. Jos jokin tuntuu vaikealta, kirjoita ylös mikä ja miksi. Ellet mitenkään saa tehtävää suoritettua, ota minuun yhteyttä niin yritän vinkata oikeaan suuntaan.

### 1. Tutustuminen

Kirjaudu sisään järjestelmään. Varsinaista käyttäjätunnusta ja salasanaa ei ole käytössä, joten klikkaa vain Kirjaudu-nappia. Tai kirjoita käyttäjätunnukseksi ja salasanaksi mitä haluat. Tiedot eivät tallennu mihinkään ainakaan ACMS:n takia.

Katsele ympärillesi hetki. Mitä linkkiotsikot tuovat mieleesi? Mitä arvelet niistä tapahtuvan? Älä klikkaa vielä mitään, kirjaa vain ajatuksesi muistiinpanoiksi.

## 2. Lähdetään liikkeelle

Yritä saada sivuston perustiedot ja rakenne muistuttamaan omaa sivustoasi. Kirjaa edistymisesi ja epäonnistumisesi.

Voit käydä tarkastamassa työsi tulokset ”julkiselta” sivustolta, jolle löytyy linkki kirjautumissivulta jos satut muuten hukkaamaan sen. Muutokset päivittyvät sinne. Ne tosin eivät tallennu, joten jos suljet selaimen tai poistut sivulta, tiedot katoavat ja joudut aloittamaan testausoperaatiosi alusta, mikäli tahdot edetä siinä.

## 3. Linkkilistat

Luo sivustolle oman sivustosi linkkilista tai ainakin samankaltainen.

## 4. Galleriat

Luo sivustolle oma kuvagalleriasi. Muista, että kaikki tapahtuu omalla koneellasi, joten kukaan ei ole varastamassa kuviasi vaikka käyttäisitkin niitä ACMS:n testauksessa.

## 5. Ulkoasut

Vaikka kokonaisen ulkoasun luominen pelkkää testausta varten vaikuttaakin hieman yliampuvalta, on tätäkin osaa varmaankin aiheellista hieman kokeilla. Olen luonut muutaman vaihtoehtoisen ulkoasun ACMS-sivustolle. Kokeilepa, osaatko muuttaa sivuston ulkoasua tarjolla olevien ominaisuuksien avulla.

Sivuston ulkoasuja ei sinänsä tarvitse kommentoida. En ole testaamassa graafisia suunnittelutaitojani tai niiden puutetta. Ylläpitoliittymän kommentointi kylläkin otetaan kiitoksella vastaan. Älä turhaan pelkää antaa negatiivista palautetta. En loukkaannu.

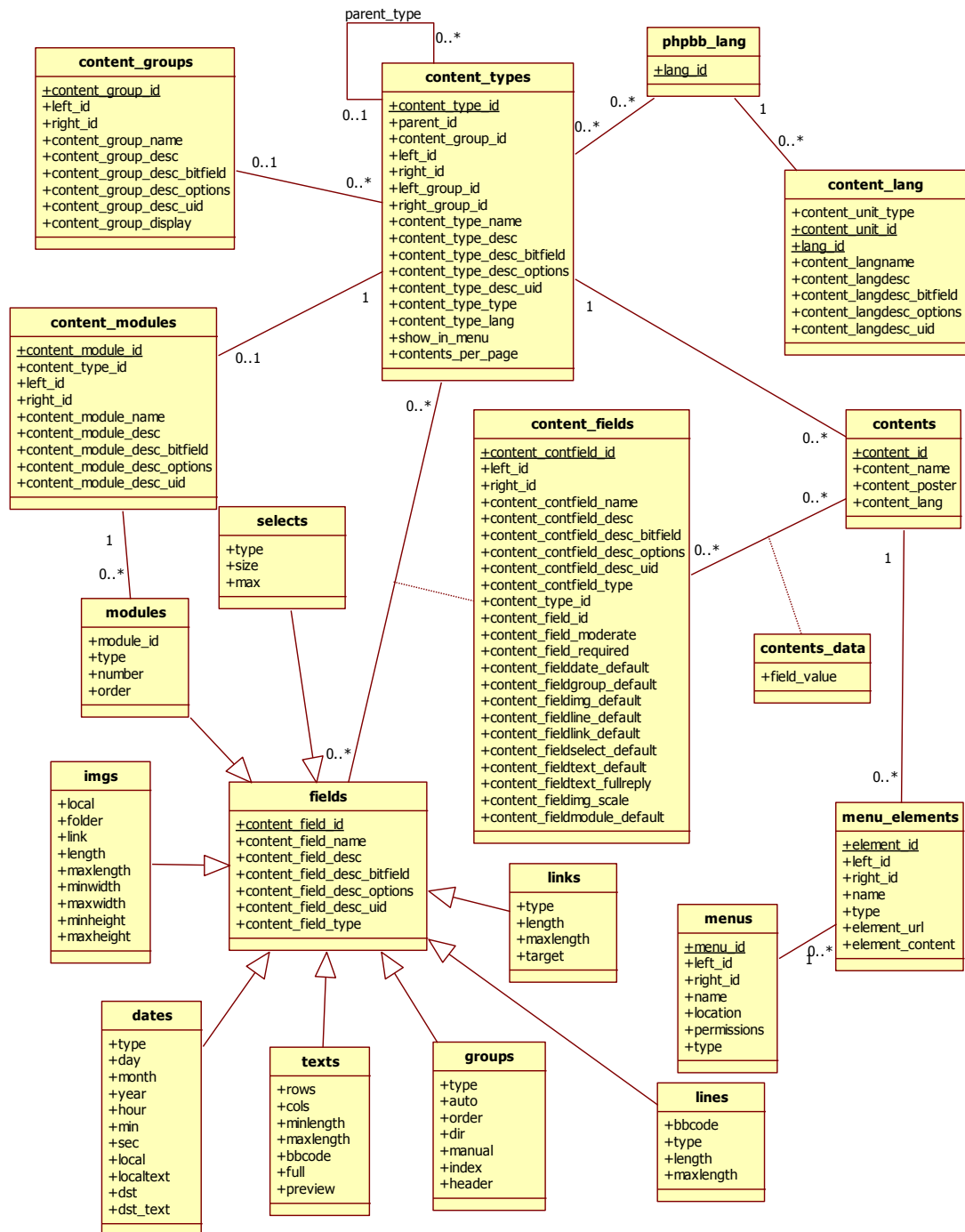
6. Se siitä

Testi on valmis. Kirjaa raportti ja palauta se minulle. Voit poistaa ACMS:n koneellasi, mikäli tahdot. Olisin iloinen ellet ainakaan jakaisi sitä omanasi. Kiitos avustasi!

TOISEN JÄRJESTELMÄN TIETOKANTA

Alleviivatut kentät ovat pääavaimia.

Taulu phpbb\_lang on phpBB:n alkuperäisiä tauluja, joten siitä en merkinnyt enempää tietoja kuin pääavaimen, jota käytän viiteavaimissa.



## TOISEN JÄRJESTELMÄN KÄYTTÖLIITTYMÄ

phpBB®  
creating communities

Administration Control Panel  
Admin index • Board index

GENERAL FORUMS POSTING USERS AND GROUPS PERMISSIONS STYLES MAINTENANCE SYSTEM **ACMS**

You are logged in as: **Admin**  
[ Logout ] [ ACP Logout ]

**ACMS**

- Yleiskatsaus**
  - Yleiset asetukset
- SISÄLLÖT**
  - Sisältöryhmät
  - Sisältötyypit
  - Sisältökentät
  - Moduulit
- VALIKOT**
  - Valikot
  - Valikkoelementit
- PALIKAT**
  - Sijainnit
  - Palikkaelementit
- OIKEUDET**
  - Sisältötyyppien oikeudet
  - Content roles
  - Content moderator roles

**ACMS**

**ACMS tilastot**

TILASTO	ARVO	TILASTO	ARVO
Kategorioita:	5	Uniikit kävijät:	42
Sisältöjä:	57	Sivulataukset:	101
ACMS:n versio:	0.0.0 [ Re-Check version ]	Orpoja liitetiedostoja:	0

**Synkronoi tai nollaa tilastot**

**Nollaa uniikit kävijät**

**Nollaa sivulataukset**

**Synkronoi tilastot**  
Laskee uudelleen sisältötyyppien ja sisältöjen määrän.

Powered by phpBB® Forum Software © phpBB Group  
Time : 0.674s | 4 Queries | GZIP : Off | Memory Usage: 6.71 MiB | Explain

ACMS:n etusivu ylläpidon hallintapaneelissa sisältää tilastoja ja joitakin enemmän tai vähemmän tarpeellisia toimintoja.

You are logged in as: Admin [ Logout ] [ ACP Logout ]

**ACMS**

Täällä voit säätää ACMS:n yleisiä ja oletusasetuksia. Nämä asetukset vaikuttavat kaikkialla ACMS:ssä.

**Sivuston yleiset asetukset**

**ACMS käytössä**  
Tällä valinnalla voit ottaa ACMS:n käyttöön ja pois käytöstä. Mikäli järjestelmä on pois käytöstä, mikään ACMS:n osa ei näy käyttäjälle. Asetukset ja sisällöt eivät kuitenkaan katoa, ellei poista järjestelmän asennusta.

Kyllä  Ei

**Oletusasetukset**

**Näytä sisältöryhmät:**  
Jos tämä on käytössä, uusi sisältöryhmä oletusarvoisesti näytetään. Voit silti valita sisältöryhmän asetuksissa olla näyttämättä sitä. Huomaa, että tämä koskee vain uusia sisältöryhmiä, olemassaolevien arvot eivät muutu.

Kyllä  Ei

**Lisää uudet sisältötyypit valikkoon:**  
Jos tämä on käytössä, uusi sisältötyyppi lisätään oletusarvoisesti valikkoon. Voit valita sisältötyypin kohdalla lisätäanko se valikkoon vai ei, tämä vain määrittää oletusarvon. Huomaa, että tämä koskee vain uusia sisältötyyppejä, olemassaolevien arvot eivät muutu.

Kyllä  Ei

**Sisältötyyppiä sivulla:**  
Tämä määrittää oletusarvon sille, kuinka monta sisältötyypin sisältöä näytetään yhdellä sivulla. 0 tarkoittaa, että kaikki sisällöt näytetään. Arvon voi ohittaa sisältötyypin asetuksista.

**Submit changes**

ACMS:n yleiset asetukset sisältävät yleisiä sisällönhallintajärjestelmän asetuksia ja oletusasetusten määrittämiä muille hallintasivuille. Kuvassa on vain osa asetuksista.

**Sisältöryhmien hallinta**

Täällä voit lisätä, poistaa ja muokata sisältötyyppien ryhmiä ja järjestellä niitä. Näitä ryhmiä käytetään sisältötyyppien organisointiin, mutta ne näkyvät käyttäjälle vain ylläpidon ja valvojen hallintapaneelissa. Sivuston rakenne määritellään itse sisältötyypeillä.

**ACMS**

**Ryhmä**

Sisältötyyppejä: 0 / Sisältöjä: 0 / Kommentteja: 0

Sisältöryhmien, sisältötyyppien ja kenttien hallinnan perusrakenne on sama kuin keskustelufoorumin keskustelualueiden: yksiköt esitetään kuvan kaltaisessa palkissa, ja yksikön nimi toimii linkkinä sisemmälle tasolle. Oikeassa reunassa on kuvakkeita, joilla voi hallita tiettyjä toimintoja. Nuoli ylös ja nuoli alas siirtävät yksikköä ylemmäs tai alemmas luettelossa (kuvassa nuolet ovat harmaana, koska elementtejä on vain yksi, eikä sitä näin ollen voi siirtää ylös eikä alas). Vihreästä rattaasta voi muokata yksikköä, punaisesta ympyrästä jossa on rasti voi poistaa sen. Yksikön luonti tapahtuu oikeassa alakulmassa olevasta ”Luo”-painikkeesta.

## Muokkaa sisältöryhmää :: Ryhmä

« Back

Täällä voit asettaa sisältöryhmälle ominaisia asetuksia.

### General options

#### Sisältöryhmän nimi:

Tämä nimi on käytössä ylläpidon asetuksissa. Nimi, jolla sisältöryhmä näkyy käyttäjille, määritetään kielikohtaisesti sisältöryhmän luonnin jälkeen. Mikäli kielikohtaista nimeä ei ole asetettu, tämä kuitenkin näytetään ryhmän nimenä.

Ryhmä

#### Sisältöryhmän kuvaus:

Tämä kuvaus on käytössä ylläpidon asetuksissa. Kuvaus, jolla sisältöryhmä näkyy käyttäjille, määritetään kielikohtaisesti sisältöryhmän luonnin jälkeen. Mikäli kielikohtaista kuvausta ei ole asetettu, tämä kuitenkin näytetään ryhmän kuvauksena.

Parse BBCode  Parse smilies  Parse links

Yes  No

#### Sisältöryhmä näytetään:

Jos sisältöryhmää ei näytetä, vain ylläpito voi nähdä sen ylläpidon asetuksista. Tämä on käytännöllistä niille sisältötyypeille, jotka ovat kategorioita, ja niille, joita ei ole tarpeen näyttää valvojen tai käyttäjien hallintapaneelissa (esimerkiksi sivut, joita ei voi kommentoida ja joita vain ylläpito voi luoda ylläpidon asetusten kautta). Itse sisällön näyttäminen ei ole riippuvainen tästä arvosta, vaan se määritellään sisältötyypin oikeuksissa.

## Muokkaa sisältötyyppiä :: Juu

« Back

Täällä voit asettaa sisältötyypille ominaisia asetuksia ja sen sisältökenttiä.

### General options

#### Sisältötyypin tyyppi:

Sisältötyypin tyyppi määrittää sen, miten sisältötyyppi esitetään ja kuinka se käyttäytyy.

Sisältö

#### Sisältöryhmä:

Mihin sisältöryhmään sisältötyyppi kuuluu. Sisältöryhmiä käytetään sisältötyyppien organisointiin, mutta pakollisia ne eivät ole. Käyttäjien ja valvojen hallintapaneelissa näkyvät kuitenkin vain ne sisältötyypit, jotka kuuluvat johonkin sisältöryhmään.

Ei ryhmää

#### Isäntätyyppi:

Isäntänä toimiva sisältötyyppi. Vaikka isäntien käyttö ei olekaan välttämätöntä, ne helpottavat monimutkaisempien sivustorakenteiden ja sisältötyyppien luontia.

Ei isäntää

#### Sisältötyypin nimi:

Tämä nimi on käytössä ylläpidon asetuksissa. Nimi, jolla sisältötyyppi näkyy käyttäjille, määritetään kielikohtaisesti sisältötyypin luonnin jälkeen. Mikäli kielikohtaista nimeä ei ole asetettu, tämä kuitenkin näytetään sisältötyypin nimenä.

Juu

#### Sisältötyypin kuvaus:

Tämä kuvaus on käytössä ylläpidon asetuksissa. Kuvaus, jolla sisältötyyppi näkyy käyttäjille, määritetään kielikohtaisesti sisältötyypin luonnin jälkeen. Mikäli kielikohtaista kuvausta ei ole asetettu, tämä kuitenkin näytetään sisältötyypin kuvauksena.

Parse BBCode  Parse smilies  Parse links

#### Sisältötyypin kieli:

Sisältötyyppi näytetään näillä kielillä. Kielikohtaiset nimet ja kuvaukset määritetään vain niille kielille, jotka on valittu tässä.

**Huomautus 1:** Jos et valitse yhtään kieltä, sisältötyyppi on nähtävissä vain ylläpidon asetuksissa.

**Huomautus 2:** Jos asennat uusia kieliä, kaikki uudella kielellä nähtävissä olevat sisältötyypit on käytävä konfiguroimassa kyseiselle kielelle. Tämä ei tapahdu automaattisesti.

British English (British English)  
Finnish (Suomi)

Valitse kaikki kielet

#### Näytä valikossa:

Käytetäänkö sisältötyyppiä sivuston valikon rakentamiseen. Jos sisältötyyppiä ei näytetä valikossa millään tasolla, sinun on tarjottava linkki siihen jollakin muulla tavalla, mikäli tahdot käyttäjien pääsevän käsiksi sen sisältöön.

Yes  No

#### Sisältöjä sivulla:

Kuinka monta sisältötyypin edustajaa näytetään yhdellä sivulla. 0 (nolla) ottaa sivutuksen pois käytöstä (kaikki sisällöt näytetään, oli niitä kuinka monta tahansa). Oletus on 15.

15

Sisältöyksikön peruselementit ovat kaikille samat, mutta eri elementtityypeillä on erilaisia lisäasetuksia, joillakin enemmän ja joillakin vähemmän. Elementeille, joilla on valittavana erilaisia tyyppejä ja tyypeillä on erilaisia asetuksia, käytetään JavaScriptiä niiden asetusten piilottamiseen, jotka eivät ole käytössä kyseisessä tyyppissä.



## Lisää kenttä :: (ei nimeä annettu)

« Back

Tässä voit määrittää sisältökentän asetuksia. Sisältökentät ovat niitä, jotka muodostavat varsinaiset sisältötyyppeihin syötettävät tiedot. Valittavissasi olevat asetukset riippuvat siitä, mikä on sisältökentän tyyppi.

### General options

#### Kentän tyyppi:

Kentän tyyppi määrittää, minkälainen arvo kenttään talletetaan ja miten sitä käsitellään.

Teksti

#### Kentän nimi:

Tätä nimeä käytetään ylläpidon asetuksissa. Kun kenttä sijoitetaan sisältötyyppiin, sille määritetään nimi sisältötyyppi- ja kieliakohtaisesti. Yksi kenttä voi kuulua useampaan kuin yhteen sisältötyyppiin, jotta samanlaisia kenttiä ei tarvitse määritellä monia. Mikäli sisältötyyppi- ja kieliakohtaista nimeä ei ole asetettu, tämä kuitenkin näytetään kentän nimenä.

#### Kentän kuvaus:

Tätä kuvausta käytetään ylläpidon asetuksissa. Samoin kuin kentän nimi, myös kentän kuvaus määritetään sisältötyyppi- ja kieliakohtaisesti. Mikäli sisältötyyppi- ja kieliakohtaista kuvausta ei ole asetettu, tämä kuitenkin näytetään kentän kuvauksena.

Parse BBCode  Parse smilies  Parse links

Yes  No

#### Pakollinen kenttä:

Jos kenttä on pakollinen, siinä ei sallita tyhjiä arvoja. Jos se taas ei ole pakollinen, eikä sisällön lähettäjä ole täyttänyt sitä, sitä ei myöskään näytetä.

Yes  No

#### Valvottava kenttä:

Jos kenttä on valvottava, muutokset siinä ilmoitetaan valvojille. Muuttunut sisältö voidaan julkaista vasta, kun muutokset on hyväksytty.

### Pitkän tekstin asetukset

#### Rivit:

Tämä arvo määrittää, kuinka korkea tekstilaatikko on. Nolla käyttää selaimen (tai tyylin) oletusta.

0

#### Merkkejä rivillä:

Tämä arvo määrittää, kuinka leveä tekstilaatikko on. Nolla käyttää selaimen (tai tyylin) oletusta.

0

#### Minimipituus:

Tämä arvo määrittää, kuinka pitkä teksti voi minimissaan olla.

0

#### Maksimipituus:

Tämä arvo määrittää, kuinka pitkä teksti voi maksimissaan olla.

0

#### Salli BBCode:

Voidaanko tekstialueessa käyttää BBCode-muotoiluja vai ei.

Yes  No

#### Koko teksti:

Näytetäänkö teksti kokonaan, vai näytetäänkö siitä vain osa, kuten blogiviesteissä. Jos arvossa Ei, näytetään *Lue lisää* -linkki, mikäli teksti on pidempi kuin näytettävä osa.

Yes  No

#### Näytettävät merkit:

Kuinka monta merkkiä näytetään ennen *Lue lisää* -linkkiä, jos *Koko teksti* on arvossa Ei. Mikäli arvo on nolla, näytetään pelkkä linkki.

0

#### Pikavastaustyyli:

Jos pikavastaustyyli on käytössä, tekstilaatikko esitetään samaan tapaan kuin pikavastaus viestiketjussa (siis ilman BBCode-nappeja). Jos pikavastaus ei ole käytössä, tekstilaatikko esitetään samaan tapaan kuin täysi viestinkirjoitustekstilaatikko.

Yes  No

Submit

Reset

Myös olosuhteet vaikuttavat siihen, mitä näytetään. ACMS mahdollistaa esimerkiksi kenttien lisäämisen sekä yksittäisinä kenttinä että suoraan sisältötyyppeihin. Kun esimerkiksi kenttää lisätään sisältötyyppiin, kuten kuvassa, tarvitaan kuitenkin enemmän tietoja kuin silloin, kun lisätään pelkkä kenttä ilman sisältötyyppejä, joten valintojakin on näkyvissä enemmän kuin pelkän sisältökentän luontinäkyvässä.