



# SOVELLUSKEHITYS DRUPAL-YMPÄRISTÖSSÄ

Hannu-Pekka Heinäjärvi

Opinnäytetyö  
Helmikuu 2011  
Tietojenkäsittelyn koulutusohjelma  
Ohjelmistotuotannon suuntautumis-  
vaihtoehto  
Tampereen ammattikorkeakoulu

TAMPEREEN AMMATTIKORKEAKOULU

Tampere University of Applied Sciences

## TIIVISTELMÄ

Tampereen ammattikorkeakoulu  
Tietojenkäsittelyn koulutusohjelma  
Ohjelmistotuotannon suuntautumisvaihtoehto

HEINÄJÄRVI, HANNU-PEKKA: Sovelluskehitys Drupal-ympäristössä

Opinnäytetyö 53 s., liitteet 2 s.  
Helmikuu 2011

---

Työskentelen tällä hetkellä yrityksessä, jossa teen töitä Drupal-ympäristössä. Alkujaan työharjoittelun kautta sain perusosaamisen Drupal-kehitykseen. Drupalin syvällisempää haltuunottoa oli sittemmin vaikea sijoittaa työ- ja opiskeluaikatauluun. Niinpä asiakkaan kanssa sovittiin, että toteutan heille Drupal-moduulin tapahtumien hallintaan. Näin saadaan yhdistettyä kaksi tavoitetta: omien taitojen kehittäminen ja asiakkaan ongelman ratkaisu.

Opinnäytetyön käytännön toteutusosan tarkoitus oli suunnitella ja toteuttaa Drupal-moduuli, joka vastaa asiakkaan tarpeita. Tämä opinnäytetyöraportti selittää opinnäytetyön toteutusosaa esimerkkinä käyttäen perusteet sovelluskehityksestä, kun alustana toimii sisällönhallintajärjestelmä Drupal. Asiaan tutustumattomat Drupal-kehittäjät saavat tästä työstä ja raportista tarvittavia apuja.

Lähteinä sekä suunnittelussa että raportissa käytettiin yhteisön Internet-sivuja, alan kirjallisuutta sekä kirjoittajan omia kokemuksia.

## ABSTRACT

Tampereen ammattikorkeakoulu  
Tampere University of Applied Sciences  
Degree Programme in Business Information Systems  
Option of Software Development

HEINÄJÄRVI, HANNU-PEKKA: Software Development in Drupal Environment

Bachelor's thesis 53 pages, appendices 2 pages.  
February 2011

---

At the moment I am working in a company where I mainly work with Drupal environment. The job that began as an internship and now continues as the job provided me the basic skills for Drupal development. It proved difficult to fit in more in-depth familiarization with Drupal into the work and study schedule, so a contract was made with the customer for an event management module done on Drupal. This combines two objectives: self-improvement and meeting the customer's needs.

The implementation part of the thesis was to design and implement a Drupal module that meets the customer's needs. The thesis explains the basics of Drupal development using the implementation part as an example. Even uninitiated Drupal developers will benefit from the thesis.

Sources for the implementation part and the thesis mainly consist of Drupal's community website and literature. Also, the writer's own experience from work plays an important role as a source material.

---

Key words: Content Management Systems, Drupal, LAMP

## SISÄLLYS

KÄSITTEITÄ.....	5
1 JOHDANTO.....	7
2 DRUPAL.....	10
2.1 Mikä on Drupal ja missä sitä käytetään.....	10
2.2 Järjestelmävaatimukset.....	10
2.3 Asentaminen Linux-ympäristössä.....	11
2.4 Single site vai multi site?.....	14
2.5 Konfigurointi.....	15
2.5.1 Blokit.....	15
2.5.2 Valikot.....	16
2.5.3 Roolit.....	16
2.5.4 Käyttäjäoikeudet.....	17
3 DRUPAL-KEHITYS YLEISESTI.....	19
3.1 Esitietovaatimukset.....	19
3.2 Koodirekisteri.....	19
3.3 Tiedostorakenne.....	19
3.4 Koukkufunktiot.....	22
3.5 Valikkojärjestelmä.....	22
3.6 Oikeudet.....	24
3.7 Lokalisointi.....	25
4 DRUPALIN RAJAPINNAT.....	28
4.1 Tietokannat.....	28
4.2 Skeema.....	30
4.2.1 Tietokannan asennus.....	31
4.2.2 Tietokannan päivitykset.....	32
4.3 Lomakkeet.....	33
4.3.1 Lomakkeiden luominen.....	33
4.3.2 Lomakkeiden validointi.....	34
5 APUVÄLINEITÄ KEHITYKSEEN.....	35
5.1 Coder-moduuli.....	35
5.2 Devel-moduuli.....	36
6 XEVENTS-MODUULISSA KÄYTETYT MENETELMÄT.....	37
6.1 Versiohallinta.....	37
6.2 Tietokanta.....	37
6.3 Lähdekoodi.....	38
6.3.1 Luokat.....	39
6.3.2 Funktiotiedostot.....	39
6.3.3 Hakemistorakenne.....	40
6.4 Testaus.....	40
6.5 Kuvia xEventsistä.....	41
7 JOHTOPÄÄTÖKSET JA POHDINTA.....	48
LÄHTEET.....	50

## KÄSITTEITÄ

AJAX	<i>Asynchronous JavaScript And XML</i> , joukko tekniikoita, joilla osa web-sovelluksesta voidaan ladata uudestaan lataamatta koko sivua.
Apache	WWW-palvelinohjelmisto, jolla WWW-sivut voidaan tarjoilla selaimille.
Blokki	"Block", Drupalin sivupohjan osa, joka näyttää tiettyä sisältöä.
CMS	<i>Content Management System</i> , järjestelmä, jolla esimerkiksi yrityksen web-sivut voidaan pitää helposti hallittavassa muodossa.
Funktio	Sovelluksen toimintoja voidaan jakaa aliohjelmiin, funktioihin. Funktioita voidaan suorittaa eri puolilta ohjelmakoodia tarvitsematta kirjoittaa samaa koodia useampaan kertaan.
GPL	<i>GNU General Public License</i> , ohjelmistojen julkaisemiseen tarkoitettu lisenssi.
jQuery	JavaScript-kirjasto, joka on suunniteltu mahdollisimman selainyhteensopivaksi.
Koukkufunktiot	"Hook", Drupalin funktioita, jotka ajetaan tietyssä tilanteessa, esimerkiksi kirjaututtaessa sisään.
LAMP	<i>Linux, Apache, MySQL &amp; PHP</i> , paketti, joka sisältää nämä ohjelmistot.

Moduuli	Drupalin toimintaa laajentava lisäosa.
MySQL	Tietokantapalvelin.
PHP	<i>PHP: Hypertext Preprocessor</i> , ohjelmointikieli dynaamisten web-sivujen kehitykseen.
SVN	<i>Subversion</i> , versionhallintajärjestelmä, jossa tiedostojen muutoksista pidetään kirjaa.

## 1 JOHDANTO

Työskentelen tällä hetkellä yrityksessä, jossa työtehtäväni koostuvat Drupal-julkaisujärjestelmään tehtävistä ohjelmointitöistä. Työkiireiden ohessa näiden menetelmien tarkempaan opiskeluun ei aikaa juuri jää. Opinnäytetyön aihetta miettiessäni pohdin, voisiko tämän opiskelun yhdistää opinnäytetyöhöni jollain tavalla. Tuttavani kanssa tuli puheeksi, että yhdistyksemme Badlaa Ry kaipaisi jonkinlaista tapahtumarekisteriä, jolla voitaisiin pitää kirjaa yhdistyksen järjestämistä tapahtumista.

Koska perusosaaminen Drupal-kehityksestä oli itselläni jo hallussa, ehdotin, että tekisin yhdistykselle Drupaliin laajennusmoduulin, jolla tämä rekisteri saataisiin toteutettua. Näin saatiin ratkaistua kaksi ongelmaa: henkilökohtainen kehittämiseni sekä yhdistyksen tarve tälle järjestelmälle.

Opinnäytetyön kirjoittamisen hetkellä Drupalin virallinen julkaisuversio oli 6.19. Versio 7.0 oli beta1-vaiheessa. Virallinen versio 7.0 julkaistaneen syksyllä 2010. (Drupal 7 Release Date 2010) Tästä syystä opinnäytetyö toteutettiin epävirallisella versiolla 7.0-beta1.

### Tarkoitus ja tavoite

Opinnäytetyön käytännön toteutusosan tarkoituksena oli toteuttaa Badlaa Ry:n käyttöön Drupaliin pohjautuva tapahtumarekisteri, xEvents-moduuli, jossa käyttäjien autentikointi hoidetaan Drupalin omin keinoin.

Projektin tavoitteena oli saada asiakkaalle käyttöön toimiva järjestelmä, jolla heidän tarpeensa saadaan täytettyä. Tavoitteena pidettiin myös sitä, että sovelluksen koodi vastaa Drupalin ohjelmointikäytäntöä. Tämän ansiosta sovellus mahdollisesti tulevaisuudessa voidaan laittaa julkiseen jakeluun yhteisö sivuille.

Tämän raportin tavoite on selkeyttää lukijalle Drupalin moduulikehitystä sekä kuvata Badlaa Ry:n käyttöön suunnitellun ja toteutetun Drupal-moduulin kehitys.

## Lähteet

Lähteenä tässä opinnäytetyössä käytettiin alan kirjallisuutta, yhteisösivuja Internetissä sekä työn tekijän omaa osaamista ja kokemuksia. Alkuperäisessä suunnitelmassa oli käyttää lähdeoteoksina useampaa kirjaa, mutta näiden kirjojen hankinta koitui ongelmalliseksi. Siispä lähdeoteosten määrä supistui yhteen.

Näistä kirjoista ensimmäinen, Pro Drupal Development, käsittelee virallisesti Drupalin 6-versiota, mutta sisältää yhteneväistä tietoa uusimpaan versioon. Kirjan kirjoittaja John VanDyk on ollut kehittämässä Drupalia vuosina 2005 – 2007 ja on muutenkin aktiivinen yhteisössä. Täten voitaneen olettaa, että Pro Drupal Development sisältää pääosin luotettavaa tietoa. Jälkimmäinen kirja, Pro Drupal 7 Development, tilattiin pyynnöstäni Tampereen ammattikorkeakoulun kirjastoon ja se käsittelee versiota 7. Tämä kirja ei kuitenkaan saapunut ajoissa, jotta sitä olisi voitu tässä opinnäytetyön toteutusosassa käyttää.

Verkkolähteistä huomattavin oli Drupalin yhteisösivusto osoitteessa <http://drupal.org>. Tästä lähteestä saatuun tietoon voidaan suhtautua melko luottavaisesti, mutta silti lähdekritiikkiä harjoittaen. Sivustolle kirjoittavat ihmiset ovat alan asiantuntijoita ja kirjoituksiin tulee korjaukset yhteisöstä, mikäli ne korjausta kaipaavat.

## Rajaukset

Opinnäytetyön aihe rajattiin moduulikehitykseen. Lisäksi opinnäytetyön käytännön toteutukseen kuului Drupal-järjestelmän asennus asiakkaan palvelimelle. Vaikkakaan se ei varsinaisesti opinnäytetyön käytännön toteutusosan aiheeseen kuulu, on se silti oleellinen osa lopputuotetta.

Teemat sekä niiden tekeminen on rajattu pois tästä opinnäytetyön aiheesta.

### Opinnäytetyöraportin rakenne

Tässä raportissa kerrotaan aluksi Drupalista yleisellä tasolla. Samassa yhteydessä käydään lyhyesti läpi Drupalin asentaminen Linux-ympäristössä sekä esitellään hyvin suppeasti perusasetukset ja niiden muokkaaminen.

Tämän jälkeen kerrotaan yleisiä asioita sovelluskehityksestä Drupalille, muun muassa esitietovaatimuksista, tiedostorakenteista sekä oleellisista komponenteista Drupalin rajapinnassa.

Yleistä asiaa seuraa yksityiskohtaisempi tutustuminen Drupalin rajapintoihin. Käyttäen omaa opinnäytetyöni käytännön toteutusta esimerkkinä selitetään, kuinka rajapintoja käytetään esimerkiksi lomakkeiden käsittelyyn.

Seuraavaksi lyhyt luku kehityksessä avuksi olevista työkaluista. Lopuksi kerrotaan opinnäytetyöni käytännön toteutusosassa käytetyistä menetelmistä. Niihin kuuluvat muun muassa käytetyt työvälineet, kuvaus tietokannan rakenteesta sekä lähdekoodin luokkatiedostot ja ohjelmointikäytännöt tarkemmin selitettynä.

## 2 DRUPAL

Tässä luvussa kerrotaan lyhyesti Drupalista, sen järjestelmävaatimuksista, asennuksesta sekä konfiguroinnista.

### 2.1 Mikä on Drupal ja missä sitä käytetään

Drupal on avoimeen lähdekoodiin perustuva sisällönhallintajärjestelmä. Se sopii hyvin sekä henkilökohtaiseen että suuryrityskäyttöön toimintojensa ja erittäin laajan laajennusmoduulivalikoimansa ansiosta. Drupalia ylläpidetään ja kehitetään yhteisöllisesti GPL-lisenssin alaisena.

Drupal sopii käytettäväksi muun muassa yhteisön portaalina, uutis-, keskustelu- ja yrityssivustona sekä verkkokauppana. Suomessa Drupalia käyttävät esimerkiksi Suomi24, Yle, Nelonen ja Pirkka.

### 2.2 Järjestelmävaatimukset

Näissä vaatimuksissa keskitytään lähinnä Drupalin jakeluversion 7 vaatimuksiin, koska käytännön toteutusosa tehtiin tällä versiolla.

WWW-palvelimeksi suositellaan Apachea käyttöjärjestelmästä riippumatta. Tämä siksi, että valtaosa Drupalin kehityksestä tehdään juuri Apache-palvelimilla, joten suurin osa testauksesta suoritetaan myös sillä. Drupal voidaan asentaa myös Microsoftin IIS-palvelimelle. (System requirements 2010)

Tietokantapalvelimeksi sopii MySQL ja vastaavat, esimerkiksi PostgreSQL tai SQLite. Näistä kuitenkin suositellaan MySQL:ää. Alkaen versiosta 7, vaatii Drupal MySQL:n version 5.0.15 tai korkeamman. (System requirements 2010)

Tietokantaan on oltava vähintään seuraavat oikeudet: select, insert, update, delete, create, drop, index ja alter. Lisäksi jotkut lisämoduulit saattavat vaatia yli-

määräisiä oikeuksia. Mikäli tietokantaan annetaan käyttäjälle kaikki oikeudet, ei näistä tarvitse huolehtia. (System requirements 2010)

PHP-versioksi suositellaan 5.2.x. Se on myös vaatimus, mikäli käytetään Drupalin versiota 7. PHP:n muistiraja (`memory_limit`) on oltava vähintään 40MT, mutta mikäli käytössä on paljon laajennusmoduuleita, voi tätä rajaa olla tarve nostaa. (System requirements 2010)

PHP:n laajennus tietokantayhteyksille (`mysql`, `mysqli`, `pgsql`) on käytettävän tietokannan mukaan asennettava ja kytkettävä päälle. Lisäksi PHP Data Objects (PDO) on oltava aktivoituna. (System requirements 2010)

Kuvankäsittelylle tarvitaan jokin PHP-kirjasto. Yleisesti käytetään GD:tä, joka on PHP:n mukana versiosta 4.3 lähtien ja yleensä oletuksena päällä. (System requirements 2010)

### 2.3 Asentaminen Linux-ympäristössä

Vaikka Drupalin asentaminen Linuxiin ei olekaan vaikeaa, olisi käyttäjällä silti hyvä olla perusosaaminen sekä Linuxin että MySQL:n komentorivikäytöstä. Lisäksi on mahdollista, että laajempaa osaamista tarvitaan, mikäli jotkin Drupalin järjestelmävaatimuksista eivät täyty.

Asennus aloitetaan hakemalla haluttu Drupalin jakeluversio palvelimelle sellaiseen hakemistoon, johon on WWW-palvelimen kautta pääsy:

```
cd ~/public_html
wget http://ftp.drupal.org/files/projects/drupal-7.0-beta1.tar.gz
```

Tämän jälkeen paketti puretaan ja asennuskansio nimetään halutun nimiseksi:

```
tar -zxf drupal-7.0-beta1.tar.gz
mv drupal-7.0-beta1 drupal
```

Mikäli tietokanta, johon Drupal asennetaan, on jo olemassa, voidaan tämä kohta jättää väliin. Uusi tietokanta luodaan seuraavasti:

```
mysqladmin -u root -p create tietokannan_nimi
```

Tämän jälkeen kysytään MySQL-palvelimen pääkäyttäjän salasanaa. Tietokantaa ei välttämättä tarvitse luoda root-käyttäjällä, mikäli jollain toisella käyttäjällä on myös oikeudet luoda tietokantoja.

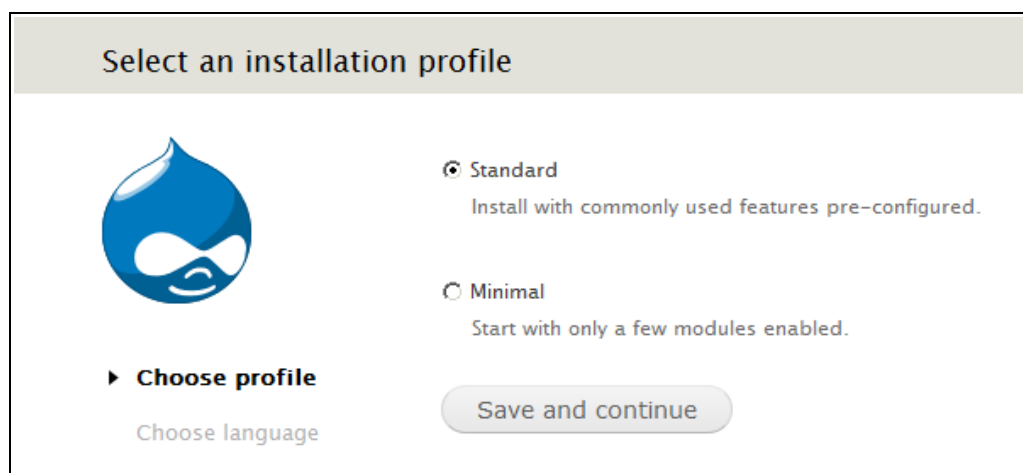
Tietokantaan täytyy vielä antaa oikeudet:

```
mysql -u root -p tietokannan_nimi
```

Jälleen kysytään pääkäyttäjän salasanaa, jonka jälkeen MySQL-palvelimen konsoli avautuu. Oikeudet tietylle käyttäjälle voidaan määrittää seuraavasti:

```
GRANT ALL ON tietokannan_nimi.* TO 'käyttäjä'@'localhost' IDENTIFIED BY 'salasana';  
FLUSH PRIVILEGES;
```

Seuraavaksi siirrytään selaimella asennusosoitteeseen [http://domain.fi/asennuskansion\\_nimi](http://domain.fi/asennuskansion_nimi). Kuvassa 1 näkyy Drupalin asennuksen ensimmäinen sivu.



Kuva 1: Drupalin asennustyyppin valinta

Vaihtoehtoista valitaan *Standard* ja jatketaan *Save and continue* -painikkeella. Tämän jälkeen valitaan asennuskieli, joka on oletuksena englanti. Jatketaan *Save and continue* -painikkeella.

Mikäli jokin vaadituista PHP:n laajennuksista puuttuu tai palvelimen asetuksissa on muuten jotain väärin, tulostuu seuraavalla sivulla taulukko korjattavista asioista. Kuvassa 2 on korostettuna järjestelmävaatimusten puutteet.

Web server	Apache/2.2.14 (Ubuntu)
PHP	5.3.2-1ubuntu4.5
PHP register globals	Disabled
PHP extensions	Enabled
Database support	Disabled
<p>Your web server does not appear to support any common PDO database extensions. Check with your hosting provider to see if they support PDO (PHP Data Objects) and offer any databases that <a href="#">Drupal supports</a>.</p>	
PHP memory limit	128M
File system	Writable ( <i>public</i> download method)
Unicode library	PHP Mbstring Extension
Settings file	The <code>./sites/default/settings.php</code> file exists.
Settings file	The settings file is writable.
<p>Check the error messages and <a href="#">proceed with the installation</a>.</p>	

*Kuva 2: Yhteenveto asennusvaatimusten täyttymisestä*

Tässä tapauksessa PHP-tulkista puuttui tuki MySQL-tietokannalle. Tämä korjattiin asentamalla tuki ja käynnistämällä WWW-palvelin uudestaan:

```
sudo apt-get install php5-mysql
sudo /etc/init.d/apache2 restart
```

Seuraavalla sivulla syötetään aiemmin luodun tietokannan asetukset. *Advanced options* -kohdan kenttien arvoja ei tarvitse muokata, mikäli tietokantapalvelin sijaitsee samalla palvelimella kuin WWW-palvelin.

Asennus käynnistyy valitsemalla *Save and continue*. Kun se on valmis, ohjeistetaan käyttäjää poistamaan järjestelmän kirjoitusoikeudet asetustiedostolta. Tämän jälkeen annetaan muutamia Drupal-sivuston perustietoja. Kohdassa *Site maintenance account* määritellään sivuston pääkäyttäjän tunnus ja salasana. Tiedot tallennetaan painamalla *Save and continue* -painiketta. Asennus on nyt valmis.

## 2.4 Single site vai multi site?

Single site -asennuksella tarkoitetaan sitä, että jokaista sivustoa varten tehdään oma Drupal-asennus, kun taas multi sitessä yhdessä Drupalissa toimii useita sivustoja.

Multi site -asennuksessa jokaiselle asennukselle tehdään sites-kansion alle oma kansionsa, joista jokainen sisältää oman asetustiedostonsa ja omat tarvittavat teemat ja moduulit.

Multi site -asennus helpottaa ylläpitäjän työtä, sillä kun kaikki sivustot toimivat saman asennuksen alaisena, riittää ainoastaan tämän asennuksen päivittäminen pitämään kaikki sivustot ajantasaisina. Toisaalta tästä voi myös koitua ongelmia, jos jokin sivustoista käyttää sellaista moduulia, joka vaatii tietyn Drupalin version. Lisäksi, mikäli kaikkien asennuksissa toimivien sivustojen ylläpitäjä ei ole sama henkilö tai henkilö tunnetusta piiristä, saattaa tämä aiheuttaa tietoturvaongelmia: ylläpitäjän oikeuksilla on näet mahdollista suorittaa PHP-koodia missä tahansa asennuksessa. (Run multiple sites from the same code base 2010)

## 2.5 Konfigurointi

Vaikka Drupal onkin asennuksen jälkeen suoraan käytettävissä, voi sen asetuksiin olla tarpeellista tehdä joitakin muutoksia. Tässä kappaleessa käydään lyhyesti läpi tärkeimpiä näistä asetuksista käyttäen esimerkkinä opinnäytetyön käytännön toteutusosaa.

### 2.5.1 Blokit

Drupalin sivurakenne koostuu ns. blokeista ("blocks"). Blokit ovat sivupohjan osia, jotka voidaan kytkeä päälle tai pois päältä. Nämä blokit ja niiden asetukset määritellään sivuston teemassa. (VanDyk 2008, 6)

Blokkeja voidaan hallita kirjautumalla sisään admin-tunnuksella. Tämän jälkeen sivun yläpalkista valitaan linkki *Structure*. Aukeavalta sivulta valitaan *Blocks*. Täältä voidaan hallita blokkeja teemoittain. Sivuston teemat, jotka sisältävät blokkeja, löytyvät sivun oikeasta ylälaidasta välilehtinä.

Asiakkaan tuotantoasennuksessa on käytössä oletusteema *Bartik*, joten tämä teema on valittava, jotta muutokset näkyvät sivustolla oikein. Oletuksena blokeista on päällä mm. *System help*, joka on linkki käyttöohjeeseen. Muita oletusblokkeja on esimerkiksi *Powered by Drupal* ja *Search form*. Tässä tapauksessa kaikki muut blokit otettiin pois käytöstä, paitsi *Main page content*, joka sijoitettiin *Content*-alueelle sekä *Navigation*, *User login* ja *User menu*, jotka sijoitettiin tässä järjestyksessä alueelle *Sidebar first*.

Blokkeja voidaan myös konfiguroida tarkemmin, esimerkiksi näkymään vain tietyillä sivuilla. Tämä tapahtuu klikkaamalla *configure*-linkkiä blokin perästä. Ensimmäisenä sivulla voidaan määrittää blokille otsikko, joka näkyy sivulla. Tämän jälkeen voidaan määrittää jokaiselle teemalle alue, jossa tämä blokki näkyy. Lopuksi kohdasta *Visibility settings* voidaan määrittää ne sivut, sisältötyypit ja roolit, joille tämä blokki näkyy. Lisäksi voidaan valita, onko kyseinen blokki käyttäjän kustomoitavissa vai ei.

### 2.5.2 Valikot

Sivustolla liikkuminen tapahtuu pääasiassa valikoissa olevien linkkien kautta. Yleensä näitä linkkejä ei tarvitse muokata, sillä kytkemällä halutut blokit pois saadaan mahdollisesti turhat linkit pois käyttäjien näkyviltä. Mikäli linkkejä on kuitenkin tarve lisätä tai muokata, voidaan se tehdä admin-tunnuksella valitsemalla yläpalkista *Structure*. Tämän jälkeen, valitsemalla *Menus*, avautuu taulukko, jossa on lueteltuna jo olemassa olevat valikot. Haluttua valikkoa pääsee muokkaamaan klikkaamalla *List links* -linkkiä. Tällä sivulla voidaan järjestellä navigaatiolinkit haluttuun järjestykseen raahaamalla niitä, muokata niitä valitsemalla *edit* tai poistaa niitä valitsemalla *delete*. Linkistä *Add link* voidaan valikkoon lisätä uusi linkki.

Drupalissa on käytössä ns. pää- ja toissijaiset linkit. Tämä tarkoittaa sitä, että jos moduulin valikkorakenne on kaksitasoinen, näkyvät päälinkit erikseen ja näiden toissijaiset linkit näkyvät toisaalla. Päälinkit näkyvät yleensä sivuston yläreunassa ja toissijaiset alempana. Asiakkaan asennuksessa molemmat kuitenkin kytkettiin pois päältä, sillä opinnäytetyön toteutusosan valikkolinkit sijaitsevat blokissa *Navigation*, joka on näkyvissä sivuston vasemmassa reunassa. Tämä tapahtuu valitsemalla valikoiden listausnäköymästä välilehti *Settings*. Täältä valittiin *No Main links* ja *No Secondary links*.

### 2.5.3 Roolit

Drupalissa yksittäiselle käyttäjälle ei voida suoraan myöntää käyttöoikeuksia tiettyihin toimintoihin, vaan nämä oikeudet jaetaan roolien kautta. Rooli on käytännössä ryhmä, jolla on tietyt oikeudet, ja jolla niputetaan käyttäjiä yhteen.

Oletuksena Drupal-asennuksessa ovat roolit *anonymous user* kirjautumattomalle käyttäjälle, *authenticated user* kirjautuneelle käyttäjälle sekä *administrator* ylläpitäjille. Näistä kahta ensimmäistä ei voida poistaa, mutta *administrator*-oikeus voidaan, mikäli muita ylläpitäjiä pääkäyttäjän lisäksi ei tarvita. Rooleja voidaan hallita valitsemalla yläpalkista *People*. Tämän jälkeen valitaan välilehti *Permis-*

sions ja alivälilehti *Roles*.

Asiakkaan asennukseen luotiin uusi rooli tapahtumien ylläpitäjille. Tämä tapahtuu kirjoittamalla roolin nimi taulukossa alimman rivin tekstikenttään ja painamalla *Add role*. Tämän jälkeen rooli raahattiin *administrator*-roolin yläpuolelle, sillä juuri luodulla roolilla tulee olemaan suppeammat oikeudet kuin administratorilla, ja on suositeltavaa, että roolit ovat järjestyksessä, jossa ylimpänä ovat rajoitetuimmat ja alimpana kaikkein laajimmat oikeudet. Kuvassa 3 näkyvät käyttäjäroolit asiakkaan asennuksessa.

		<a href="#">Show row weights</a>
NAME	OPERATIONS	
+ anonymous user (locked)	<a href="#">edit permissions</a>	
+ authenticated user (locked)	<a href="#">edit permissions</a>	
+ xevents_admin	<a href="#">edit role</a>	<a href="#">edit permissions</a>
+ administrator	<a href="#">edit role</a>	<a href="#">edit permissions</a>

[Add role](#)

Kuva 3: Käyttäjäroolit asiakkaan järjestelmässä

#### 2.5.4 Käyttäjäoikeudet

Kuten edellä kerrottiin, ei yksittäiselle käyttäjälle voida suoraan lisätä käyttäjäoikeuksia. Sen sijaan käyttäjälle myönnetään käyttäjärooli tai useita rooleja, joilla on omat oikeutensa tehdä asioita. Rooleille myönnettävät käyttäjäoikeudet tulevat suoraan eri moduulien määrytyksistä. Nämä myönnettävät oikeudet voidaan määrittellä eri rooleille valitsemalla yläpalkista *People* ja vaihtamalla välilehdelle *Permissions*. Taulukossa näkyvät allekkain kaikki järjestelmän moduulit ja näiden määrittelemät käyttäjäoikeudet. Oikeassa reunassa näkyvät roolit, joille näitä oikeuksia voidaan myöntää. Asiakkaan asennuksessa myönnettiin roolille *xevents\_admin* käyttäjäoikeus *xEvents Admin* sekä kirjautuneelle käyttäjälle oi-

keus *xEvents User*. Tämä tapahtui selaamalla sivun loppuun, josta moduuli *xEvents* löytyy, ja valitsemalla riviltä *xEvents User* kohta *Authenticated User*, sekä riviltä *xEvents Admin* kohdat *xevents\_admin* ja *administrator*. Lopuksi painetaan *Save permissions* -painiketta, jotta muutokset tallentuvat. Kuvassa 4 näkyvät käyttäjäoikeudet asiakkaan asennuksessa.

<b>xEvents</b>				
<b>xEvents User</b>				
Allows user to view events and join them.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
<b>xEvents Admin</b>				
Allows user to administrate events, users, locations and types.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

*Kuva 4: Käyttäjäoikeudet asiakkaan järjestelmässä*

Lopuksi käyttäjille pitää vielä myöntää roolit, jotta käyttäjäoikeudet siirtyvät heille. Tämä tapahtuu valitsemalla yläpalkista *People*. Avautuvassa taulukossa näkyvät kaikki käyttäjät. Halutut käyttäjät rastitetaan, jonka jälkeen valitaan *Update options* -valikosta *Add a role to the selected users* -otsikon alta rooli, joka halutaan käyttäjille myöntää. Lopuksi painetaan *Update*-painiketta, jotta roolimääritykset tallentuvat.

## 3 DRUPAL-KEHITYS YLEISESTI

Tässä luvussa käydään yleisesti läpi sovelluskehitystä Drupalille. Aiheina ovat muun muassa moduulitiedostot, valikkojärjestelmä sekä niin sanotut koukkufunktiot.

### 3.1 Esitietovaatimukset

Vähimmäisvaatimuksena on PHP-kielen osaaminen. Periaatteessa pelkästään tämä riittää yksinkertaisen moduulin rakentamiseen. Lähes aina tarvitaan kuitenkin myös tietokantaa, joten tietokantaosaaminen jostain tuetusta tietokannasta vaaditaan.

HTML- ja CSS-aidot ovat välttämättömiä, mikäli halutaan käyttää Drupalin sivupohjia ("template"). Näillä voidaan jakaa varsinainen koodi erilleen sivun asettelumääritteistä. Lisäksi JavaScriptillä voidaan luoda lisätoiminnallisuutta moduuliin. Drupalissa tulee mukana tuki jQuery-kirjaston käyttöön.

### 3.2 Koodirekisteri

Drupalin koodirekisteri sisältää kaikki Drupalin ytimen ja aktiivisten moduulien luokat ja rajapinnat. Luokat ladataan rekisteriin automaattisesti tarvittaessa. Rekisterin ansiosta esimerkiksi omaa moduulia kehittäessä voidaan käyttää muiden moduulien ja ytimen tarjoamia toiminnallisuuksia. (Drupal's code registry 2010)

### 3.3 Tiedostorakenne

Drupal sisältää tiedostoja, jotka ovat moduuleille enemmän tai vähemmän pakollisia.

Moduulin `.install`-tiedosto sisältää toimintoja, jotka suoritetaan moduulia asennettaessa sekä asennusta poistettaessa. Yksi tällainen toiminto on esimerkiksi tietokannan asennus. Tämän opinnäytetyön toteutusosiossa tämä tiedosto asentaa moduulin tietokannan käyttämällä koukkufunktiota `hook_schema`, ja tämän jälkeen lisää tarvittavat vierasavaimet tietokantataulujen välille. Moduulin asennusta poistettaessa tämä tiedosto poistaa ensin vierasavaimet, jonka jälkeen poistetaan tietokantataulut.

Moduulin `.module`-tiedosto sisältää moduulin käyttämiä koukkufunktioita. Tällaisia koukkufunktioita ovat muun muassa `hook_menu`, joka määrittää moduulin valikkorakenteen sekä `hook_permission`, jolla voidaan luoda uusia, rooleille asetettavia käyttöoikeuksia. Opinnäytetyön toteutusosiossa tähän tiedostoon toteutettiin funktiot `xevents_permission`, `xevents_menu` ja `xevents_library`. Lisäksi tähän tiedostoon sisältyi yksinkertainen apufunktio koodin rakenteen tutkimiseen.

`.info`-tiedosto sisältää moduulin käyttämää metadataa. Tällaista dataa ovat esimerkiksi moduulin nimi, kuvaus ja sen vaatimat muut moduulit. Lähteen *Writing .info files* (2010) mukaan pakollisia tietoja ovat:

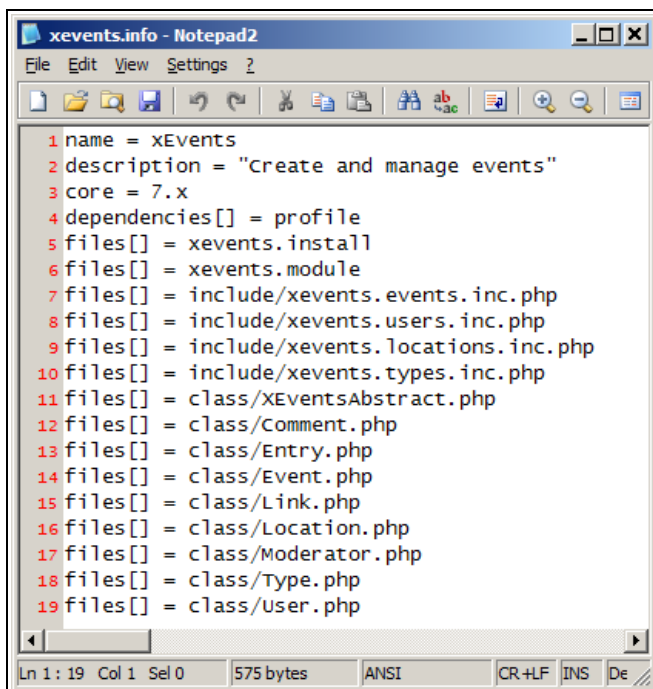
- `Name`, moduulin nimi. Tämä näkyy esimerkiksi ylläpitäjän moduulilistauksessa
- `Description`, moduulin kuvaus. Lyhyt kuvaus moduulista, joka näkyy myös moduulilistauksessa
- `Core`, Drupalin ytimen vaadittu versio. Määrittää, mille Drupalin versiolle tämä moduuli on toteutettu
- `Files[]`, moduulin kooditiedostot. Tähän taulukkoon lisätään ne tiedostot, jotka sisältävät moduulin lähdekoodia. Nämä tiedostot käydään läpi ja niiden sisältämät luokat ja funktiot lisätään Drupalin koodirekisteriin.

Lähteen *Writing .info files* (2010) mukaan vapaaehtoisia tietoja ovat:

- `Dependencies[]`, moduulin riippuvuudet. Mikäli moduuli tarvitsee toimiakseen jonkun muun moduulin, määritellään se tähän. Omaa moduulia aktivoitaessa tämä vaadittu moduuli otetaan käyttöön automaattisesti

- Package, pakkaus. Mikäli moduulisi on osa jotain isompaa kokonaisuutta, voi tällä määritteellä niputtaa niitä yhteen esimerkiksi moduulilistauksessa
- PHP, PHP:n versio. Määrittää, mikä PHP:n versio pitää palvelimella olla asennettuna, jotta tätä moduulia voidaan käyttää
- Required, vaadittu moduuli. Jos tämä asetetaan arvoon true, ei moduulia voida kytkeä pois päältä. Yleensä tällaisia ovat ytimen mukana, sivuston toimintaan olennaisesti vaikuttavat moduulit
- Hidden, piilotettu moduuli. Jos tämä asetetaan arvoon true, ei moduuli näy ylläpitäjän moduulilistassa.

Kuvassa 5 on kuvakaappaus opinnäytetyön toteutusosana tuotetun moduulin .info-tiedostosta.



```
1 name = xEvents
2 description = "Create and manage events"
3 core = 7.x
4 dependencies[] = profile
5 files[] = xevents.install
6 files[] = xevents.module
7 files[] = include/xevents.events.inc.php
8 files[] = include/xevents.users.inc.php
9 files[] = include/xevents.locations.inc.php
10 files[] = include/xevents.types.inc.php
11 files[] = class/xEventsAbstract.php
12 files[] = class/Comment.php
13 files[] = class/Entry.php
14 files[] = class/Event.php
15 files[] = class/Link.php
16 files[] = class/Location.php
17 files[] = class/Moderator.php
18 files[] = class/Type.php
19 files[] = class/User.php
```

Kuva 5: xEvents-moduulin .info-tiedosto

### 3.4 Koukkufunktiot

Drupalin koukkufunktiot ("hooks") vastaavat Drupalin sisäisiä tapahtumia (Van-Dyk 2008, 6). Järjestelmässä tapahtuvia asioita voidaan "koukuttaa" eli kytkeä näihin funktioihin. Esimerkiksi, kun käyttäjä kirjautuu sisään, laukaistaan koukkufunktio *hook\_user*. Tällöin kaikista moduuleista, jotka tämän koukkufunktion määrittävät, suoritetaan tämä funktio. Jos opinnäytetyön toteutusosassa tehtyyn moduuliin olisi määritetty funktio *xevents\_user*, suoritettaisiin se käyttäjän kirjautuessa sisään. Tähän funktioon voitaisiin toteuttaa haluttuja toiminnallisuuksia, esimerkiksi ohjata käyttäjä omalle profiilisivulleen. Koukkufunktiot nimetään aina niin, että hook-sana korvataan moduulin nimellä. Alaviivan jälkeinen loppuosa riippuu siitä, mikä koukkufunktio toteutetaan.

Muita tärkeitä koukkufunktioita, joita tämänkin opinnäytetyön toteutusosassa on käytetty, ovat esimerkiksi *hook\_menu* ja *hook\_permission*.

### 3.5 Valikkojärjestelmä

Drupalissa koukkufunktion *hook\_menu* avulla voidaan määritellä moduulin valikkorakenne sekä sivut, joilla vierailtaessa toimintoja tapahtuu. Koodiesimerkissä 1 on yksinkertainen toteutus *hook\_menu*-koukkufunktiosta.

```
function xevents_menu() {
    $items = array();

    // Events
    $items['xevents/event/%'] = array(

        // Määrittää sivun otsikon
        'title' => 'Event details',

        // Määrittää suoritettavan funktion
        'page callback' => 'drupal_get_form',

        // Määrittää kohdefunktion parametrit
        'page arguments' => array('xevents_view_event_form', 2),
```

```

// Määrittää vaaditut oikeudet
'access arguments' => array('xevents user'),

// Määrittää mistä suoritettava funktio löytyy
'file' => 'include/xevents.events.inc.php',

// Määrittää valikko-objektin tyyppin
'type' => MENU_CALLBACK,
);

return $items;
}

```

### *Koodiesimerkki 1: Yksinkertainen toteutus hook\_menu-koukkufunktiosta*

Funktiossa *\$items* on PHP-taulukko, joka sisältää valikon osat sekä osoitteet, joissa moduulin toimintoja sijaitsee. Taulukon jokainen solu on oma osoitteensa. Solut itsessään ovat myös taulukoita, jotka sisältävät jokaisen osoitteen määrittäykset. Esimerkissä *\$items['xevents/event/%']* tarkoittaa sitä, että asennuksen sivulla <http://drupal-asennuksen-osoite.net/xevents/event/1> tapahtuu jotain. Opinnäytetyön toteutusosassa tämä sivu näyttää yhden tapahtuman tiedot.

Taulukon indeksi *'page callback'* => *'drupal\_get\_form'* määrittää, mikä funktio suoritetaan, kun käyttäjä avaa tämän osoitteen. Tässä tapauksessa kutsutaan Drupalin omaa *drupal\_get\_form()*-funktioita, joka näyttää lomakkeen. Kohdefunktiolle voidaan viedä parametrejä määrittämällä *'page arguments'* -indeksi. Esimerkissä ensimmäisenä parametrinä viedään sen lomakkeen nimi, joka on tarkoitus tulostaa. Toinen parametri poimitaan suoraan osoiteriviltä, heti osoitteen osan */event/* jälkeen.

Indeksi *'type'* => *MENU\_CALLBACK* ei aseta linkkiä valikkoon, mutta rekisteröi tämän osoitteen suoritusta varten. Mikäli tälle sivulle siis saavutaan, suoritetaan kohdefunktio. Muita mahdollisia vaihtoehtoja *MENU\_CALLBACK*:lle ovat *MENU\_DEFAULT\_LOCAL\_TASK* ja *MENU\_LOCAL\_TASK*, jotka lisäävät valikolinkin sijasta uuden välilehden sivulle sekä *MENU\_NORMAL\_ITEM*, joka li-

sää näkyvän linkin valikkoon.

### 3.6 Oikeudet

Koukkufunktiolla *hook\_permission* voidaan luoda uusia oikeuksia, joita voidaan määrittää pääsyvaatimuksiksi *hook\_menu*-koukkufunktion *'access arguments'*-kohdassa. Tässä funktiossa määriteltävät oikeudet tulevat näkyviin ylläpitäjälle taulukossa, jossa voidaan määrittää eri rooleille eri oikeudet. Koodiesimerkissä 2 luodaan moduuliin uusi käyttöoikeus, jota käytetään koodissa rajoittamaan toiminnot vain tietyille osalle käyttäjistä.

```
function xevents_permission() {
    $permissions = array();

    $permissions['xevents admin'] = array(
        'title' => t('xEvents Admin'),
        'description' => t('Allows user to administrate events'),
    );

    return $permissions;
}
```

#### *Koodiesimerkki 2: Esimerkki moduulin käyttöoikeuksien määrittelystä*

Esimerkissä muuttuja *\$permissions* on PHP-taulukko, joka sisältää luotavat käyttäjäoikeudet. Jokainen taulukon solu on oma oikeutensa, solun indeksi kertoo luotavan oikeuden nimen. Kohta *'title' => t('xEvents Admin')* määrittää oikeuden otsikon, joka näkyy taulukossa, jossa rooleille määritetään käyttäjäoikeuksia. Alempi *description*-kohta määrittää kuvauksen, joka näkyy otsikon alapuolella samassa taulukossa. Molemmat näistä on sijoitettu *t()*-funktion sisään, joka tarkoittaa sitä, että nämä ovat käännettäviä tekstejä.

### 3.7 Lokalisointi

Lokalisoinnilla tarkoitetaan käyttöliittymän tekstien korvaamista käyttäjän kieltä vastaavilla termeillä. Koska Drupalia kehittää ja ylläpitää kansainvälinen yhteisö, tukee se lokalisointia vahvasti. (VanDyk 2008, 407)

Lokalisointi vaatii, että ydinmoduuli *Locale* on kytketty päälle. Ylläpitäjän oikeuksilla se tapahtuu valitsemalla yläpalkista *Modules*. Tämän jälkeen valitaan moduuli *Locale* ja painetaan sivun lopusta *Save configuration* -painiketta. Seuraavaksi kannattaa järjestelmään lisätä kohdekieli, jolle käännökset tehdään. Kieliä voidaan hallita valitsemalla yläpalkista *Configuration*, ja tämän jälkeen *Languages*. Sivun yläreunassa on linkki *Add language*. Tämä avaa lomakkeen, josta voidaan valita joko valmis kieli tai luoda kokonaan oma. Yleensä kuitenkin valmiit kielet ovat riittävät. Haluttu kieli valitaan valikosta ja painetaan *Add language* -painiketta.

Lähdekoodissa käännettävät tekstit sijoitetaan Drupalin *t()*-funktion sisään. Yksinkertaisimmillaan käännös toimii esimerkiksi seuraavasti:

```
$string = t('Welcome, user!');
```

Tämä sijoittaa muuttujaan *\$string* tekstin *Welcome, user* käännettynä järjestelmässä valitulle kielelle. Usein käännettävään tekstiin on myös tarve sisällyttää muuttujia, esimerkiksi käyttäjänimiä, joita ei kuitenkaan käännetä muun tekstin mukana. Tämä voidaan toteuttaa seuraavasti:

```
$username = 'Hannu-Pekka';  
$string = t('Welcome, !user', array('!user' => $username));
```

Tämä sijoittaa muuttujaan *\$string* tekstin *Welcome, Hannu-Pekka*. *t()*-funktiolle parametrinä annetusta tekstistä käännetään muut, paitsi *!*-merkillä erotetut osat. Nämä osat, joita ei ole tarkoitus kääntää, annetaan *t()*-funktiolle toisena parametrinä PHP-taulukkona. Jos *!*-merkki korvataan *%*-merkillä, ei tätä sanaa edelleenkään käännetä, mutta sen sijaan se sijoitetaan HTML:n *<em>*-tagien sisään, jolloin se näkyy sivulla korostettuna. (Using t-function 2010)

Jos !- ja %-merkkien sijasta käytetään @-merkkiä, ajetaan sillä merkityt osat tekstistä Drupalin *check\_plain()*-funktion läpi. Tämä funktio poistaa tulostettavasta tekstistä HTML-muotoilut:

```
$username = '<b>Hannu-Pekka</b>';
$string = t('Welcome, @user', array('@user' => $username));
```

Käännöstiedostojen luomiseen ja muokkaamiseen kannattaa käyttää erillistä Poedit-ohjelmaa (<http://www.poedit.net/>). Jotta tälle ohjelmalle saadaan koostettua tiedosto, jossa käännökset sijaitsevat, tarvitaan Drupal-asennukseen moduuli nimeltä *Translation template extractor* (<http://drupal.org/project/potx>).

Kun tämä moduuli on kytketty päälle, voidaan käännökset koostaa tiedostoon valitsemalla yläpalkista *Configuration* ja tämän jälkeen *Translate interface*. Väli-lehdeltä *Extract* valitaan haluttu moduuli, josta käännökset haetaan. Lopuksi painetaan sivun alareunasta painiketta *Extract*, jolloin moduuli tarjoaa koostettua .pot-tiedostoa.

Seuraavaksi käynnistetään asennettu Poedit ja valitaan *Tiedosto*-valikosta (File) *Uusi katalogi POT-tiedostosta*. Tässä valitaan tallennettu .pot-tiedosto. Ohjelma kysyy valinnaisia tietoja ja näiden syöttämisen jälkeen päästään OK-painiketta painamalla tallentamaan varsinainen käännöstiedosto. Kuvassa 6 on näkymä Poedit-ohjelmasta.

The screenshot shows the Poedit application window. At the top, there are labels for 'Fullname:' and 'Username:'. Below these, there is a table with two columns. The first column contains three items, each with a star icon: 'Mail:', 'Is paid', and 'My events'. The second column contains three items: 'Sähköposti:', 'Maksettu?', and 'Tapahtumani'. Below the table, there are two text input fields. The first is labeled 'Mail:' and the second is labeled 'Sähköposti:'. The interface has a classic Windows-style border and scrollbars.

Kuva 6: Kääntäminen Poedit-ohjelmalla

Kun käännöstiedosto on auki ohjelmassa, näkyy vasemmalla puolella alkuperäinen teksti ja oikealla puolella sen käännös. Valitaan käännettävä teksti ja syötetään tämän käännetty vastine alimpaan tekstilaatikkoon.

Valmis käännöstiedosto voidaan tuoda järjestelmään valitsemalla yläpalkista *Configuration* ja *Translate interface*. Tuominen tapahtuu välilehdeltä *Import*. Käännöstiedosto valitaan kohdasta *Language file*, kohdekieli valikosta *Import into* ja valitaan *Mode*-kohdasta ylempi vaihtoehto. *Import*-painikkeesta kielitiedosto tuodaan järjestelmään.

## 4 DRUPALIN RAJAPINNAT

Tässä luvussa käsitellään Drupalin tarjoamia rajapintoja. Rajapinnoilla voidaan esimerkiksi käsitellä tietokantaa tai luoda ja muokata lomakkeita.

### 4.1 Tietokannat

Drupal sisältää rajapinnan tietokannan kanssa toimimiseen. Tämän kautta voidaan tehdä erilaisia kyselyitä tietokantaan. Kyselyiden suorittaminen rajapinnan kautta mahdollistaa niiden toimimisen eri tietokantapalvelimilla sekä komentojen ketjuttamisen. (Database general concepts 2010)

Tietokantaan voidaan syöttää uutta tietoa käyttämällä funktiota *db\_insert*. Koodiesimerkissä 3 syötetään opinnäytetyön toteutusosan tietokantaan uusi linkki.

```
$lid = db_insert('xevents_link')
    ->fields(array(
        'lid' => 0,
        'uid' => 1,
        'eid' => 10,
        'url' => 'http://www.google.fi',
        'description' => 'Google'
    ))
    ->execute();
```

#### *Koodiesimerkki 3: Uuden linkin syöttäminen tietokantaan rajapinnan avulla*

Esimerkissä funktion *db\_insert* parametri on sen taulun nimi, johon tiedot tallennetaan. Koska komentoja voidaan ketjuttaa, voidaan heti perään käyttää funktiota *fields*, jolla määritetään, mihin kenttiin tallennetaan mitään tietoa. *Fields*-funktio saa parametrinaan assosiatiivisen taulukon, jonka indeksit ovat kenttien nimiä ja arvot tietokantaan tallennettavaa tietoa. Lopuksi kysely suoritetaan lisäämällä ketjuun funktio *execute*. Tämä palauttaa muuttujaan *\$lid* juuri syötetyn rivin id-numeron, mikäli tietokannassa kenttä *lid* on määritelty automaattisesti

kasvavaksi. (Insert queries 2010)

Haut tietokannasta voidaan toteuttaa funktiolla *db\_select*. Koodiesimerkissä 4 haetaan opinnäytetyön toteutusosan tietokannasta kaikkien linkkien osoitteet.

```
$result = db_select('xevents_link', 'link')
    ->fields('link', array('url'))
    ->execute()->fetchCol();
```

#### *Koodiesimerkki 4: Haetaan tallennettujen linkkien osoitteet tietokannasta*

Funktio *db\_select* saa ensimmäiseksi parametrikseen sen tietokantataulun nimen, josta tietoa ollaan hakemassa. Toinen parametri on valinnainen. Tällä voidaan määrittää vaihtoehtoinen, lyhyempi nimi taululle kyselyyn myöhempää käyttöä varten. Seuraavaksi ketjuun lisätään funktio *fields*, joka saa ensimmäisenä parametrinaan taulun nimen. Toinen parametri on taulukko, jossa luetaan ne kentät, jotka taulusta halutaan noutaa. Kysely suoritetaan funktiolla *execute*. Tässä esimerkissä loppuun lisätään vielä kutsu funktioon *fetchCol*, joka palauttaa halutun kentän yksinkertaisessa taulukossa. Mikäli käytettäisiin funktiota *fetchAll*, palautettaisiin tiedot taulukossa, jossa jokainen indeksi on oma olionsa. Olion sisältö määräytyisi valittujen tietokantakenttien mukaan. (Dynamic queries 2010)

Tietojen päivitys tietokantaan hoidetaan *db\_update*-funktiolla. Koodiesimerkissä 5 päivitetään opinnäytetyön toteutusosan tietokantaan sen linkin osoite, jonka ID on 1.

```
$updated = db_update('xevents_link')
    ->fields(array(
        'url' => 'http://www.google.com'
    ))
    ->condition('lid', 1, '=')
    ->execute();
```

#### *Koodiesimerkki 5: Päivitetään tietokantaan tallennetun linkin osoite*

*db\_update* toimii lähes samalla tavalla kuin aiemmin esitelty *db\_insert*. Parametrinä annetaan päivitettävän taulun nimi. *Fields*-funktiolle annetaan assosiativisenä taulukkona ne kentät ja arvot, jotka tahdotaan päivittää. Funktio *condition* saa ensimmäisenä parametrinaan kentän nimen. Tästä kentästä pitää löytyä arvo, joka määrittellään funktion toisena parametrinä. Esimerkissä siis päivitetään linkkiä, jonka *lid* tietokannassa on 1. Päivitys suoritetaan kutsumalla *execute*-funktiota, joka palauttaa päivitettyjen rivien lukumäärän *\$updated*-muuttujaan. (Update queries 2010)

Tietokannasta voidaan poistaa rivejä käyttämällä funktiota *db\_delete*. Koodiesimerkissä 6 poistetaan edellisessä kappaleessa päivitetty linkki.

```
$deleted = db_delete('xevents_link')
    ->condition('lid', 1, '=')
    ->execute();
```

#### *Koodiesimerkki 6: Poistetaan tietokantaan tallennettu linkki*

Funktiolle *db\_delete* annetaan parametrinä sen taulun nimi, josta rivejä poistetaan. *Condition*-funktiolle annetaan, kuten edellisessäkin esimerkissä, haettavan kentän nimi ja arvo, joka pitää löytyä kentästä. Tässä esimerkissä siis poistetaan linkki, jonka *lid* on 1. Lopuksi kysely suoritetaan funktiolla *execute*, joka palauttaa muuttujaan *\$deleted* poistettujen rivien määrän. (Delete queries 2010)

## 4.2 Skeema

Skeema-rajapinnalla ("Schema API") moduulin tietokantamäärittelyt voidaan toteuttaa PHP:n taulukkomuodossa. Ennen tätä rajapintaa kirjoitettiin jokaiselle tietokantataululle oma SQL-kyselynsä, joka vastasi taulun ja kenttien luomisesta. Tämä oli huono keino tietokannan ylläpitoon, sillä tietokannan kasvaessa ja mahdollisten rakennemuutosten määrän lisääntyessä selkeys kärsi huomattavasti. Lisäksi SQL-kyselyt luotiin aina nimenomaan tiettyä tietokantapalvelinta, esimerkiksi MySQL:ää, varten. Tämä rajoitti moduulin yhteensopivuutta muiden tietokantojen kanssa. (Schema API 2010)

#### 4.2.1 Tietokannan asennus

Tietokantataulujen määrittämiseen käytetään PHP-taulukkoa. Koodiesimerkissä 7 on esitelty yksinkertaisen tietokantataulun rakenne.

```
$schema['xevents_example'] = array(
    'fields' => array(
        'eid' => array(
            'type' => 'serial',
            'size' => 'normal',
            'not null' => TRUE),
        'uid' => array(
            'type' => 'int',
            'size' => 'normal',
            'not null' => TRUE),
        'primary key' => array('eid'),
    )
);
```

#### *Koodiesimerkki 7: Yksinkertaisen tietokantataulun rakennemäärittely*

Tässä esimerkissä luodaan taulu `xevents_example`, joka sisältää 2 kenttää:

- `Eid`, kokonaislukukenttä, jonka arvo määräytyy automaattisesti ja kasvaa joka rivillä.
- `Uid`, kokonaislukukenttä, joka ei voi olla tyhjä.

Taulukon indeksiksi määritellään luotavan taulun nimi. Tämä taulukon indeksi sisältää ennalta määritettyjä avaimia:

- `Fields`, sisältää tietokantataulun kentät sekä niiden määritteet (Schema API reference 2010):
  - `Type`, kentän tietotyyppi, esimerkissä kentän tietotyyppi *serial* vastaa MySQL:n määrittelyä `auto_increment`.
  - `Size`, kentän suurin mahdollinen arvo.
  - `Not null`, voidaanko kenttä jättää tyhjäksi.

Muita mahdollisia määritteitä ovat esimerkiksi:

- Default, kentän oletusarvo mikäli muuta ei anneta.
- Length, tietyillä tietotyypeillä kentän maksimipituus.
- Primary key, taulun pääavain.
- Unique keys, taulun uniikit kentät.
- Indexes, taulun sisältämät indeksoidut kentät.

Nämä tietokantataulujen määrytykset sijaitsevat koukkufunktiossa *hook\_schema*. Kun moduuli otetaan ensimmäistä kertaa käyttöön, suoritetaan tämä koukkufunktio ja määritellyt tietokantataulut luodaan. Drupalin versiosta 7 lähtien tätä koukkufunktiota ei tarvitse enää erikseen kutsua koukkufunktiossa *hook\_install*. (Converting 6.x modules to 7.x 2010)

Kun moduuli otetaan pois käytöstä ja asennus poistetaan, kutsutaan vastaavasti *hook\_schema* -funktiota ja siinä määritellyt tietokantataulut poistetaan.

#### 4.2.2 Tietokannan päivitykset

Mikäli moduulin tietokantarakennetta on tarve päivittää, täytyy huolehtia kahdesta asiasta:

- Nämä muutokset tulee päivittää alkuperäiseen skeemaan.
- Moduulin pakolliseen *.install*-tiedostoon tulee toteuttaa koukkufunktio *hook\_update\_n()*.

Tässä n tarkoittaa päivityksen järjestysnumeroa. Funktion rungossa toteutetaan tietokannan muutokset. Koodiesimerkissä 8 on esitelty kentän lisäys (Database functions: *db\_add\_field* 2010) johonkin tietokannan tauluun.

```
function xevents_update_1() {
    $new_field = array(
        'eid' => array(
            'type' => 'int',
```

```

        'size' => 'normal',
        'not null' => TRUE
    )
);

db_add_field('xevents_event', 'active', $new_field);
}

```

### *Koodiesimerkki 8: Kentän lisääminen tietokantatauluun*

Tämä lisää tauluun *xevents\_event*-kentän nimeltä *active*, joka on tietotyyppiltään kokonaisluku, eikä se voi olla tyhjä.

## 4.3 Lomakkeet

Drupalin lomakerajapinnalla lomakkeet voidaan määritellä PHP-taulukossa, joka sisältää lomakkeen kentät ja niiden määritteet. Tämä taulukko muunnetaan HTML-koodiksi oikeaa sivua ladattaessa. Tämän ansiosta lomaketta on helppo muokata ja mahdollisesti käyttää toisessa moduulissa. Lomakerajapinta lisää myös tietoturvaa, sillä lomakkeita käsiteltäessä tarkistetaan, että lähetetty lomake on oikea, eikä käyttäjän mahdollisesti väärentämä. (VanDyk 2008, 221)

### 4.3.1 Lomakkeiden luominen

Kuten edellä mainitaan, lomakkeiden rakenne määritellään Drupalissa PHP-taulukoon. Koodiesimerkissä 9 on yksinkertainen lomake, joka sisältää yhden tekstikentän ja painikkeen.

```

function xevents_type_add_form($form, $form_state) {
    $form['name'] = array(
        '#type' => 'textfield',
        '#title' => t('Name:'),
        '#required' => TRUE,
        '#maxlength' => 45,
    );
}

```

```

$form['submit'] = array(
  '#type' => 'submit',
  '#value' => t('Add'),
);

return $form;
}

```

### *Koodiesimerkki 9: Yksinkertaisen lomakkeen rakennemäärittely*

Tässä muuttuja `$form` on taulukko, joka sisältää luotavan lomakkeen elementit. Lomakkeen elementit määritetään assosiatiivisina taulukoina. Elementin tyyppi määritellään indeksillä `type`. Mahdollisia tyyppejä ovat normaalit HTML-kielen lomake-elementit, esimerkiksi tekstikenttä ('textfield'), alasvetovalikko ('select') ja salasanakenttä ('password'). Lisäksi tyyppiä voidaan määrittää jokin Drupalin oma lomake-elementti (Form API reference 2010). Esimerkissä tekstikentän otsikoksi on asetettu käännettävä teksti Name indeksillä `#title`. Koska elementin määritteistä löytyy myös `#required`, on tämä kenttä pakko täyttää, mutta korkeintaan 45 merkkiä sisältävällä tekstillä. Tämän määrittää `#maxlength`.

Lomakkeen viimeinen elementti on tyypiltään `submit`. Tämä lisää painikkeen, joka lähettää lomakkeen eteenpäin. Mikäli painikkeelle ei anneta `#submit`-määrittettä, käsitellään lomake oletuksena funktiossa, jonka nimi on lomakefunktion nimi + `_submit`. Esimerkkitapauksessa lomake siis käsitellään funktiossa `xevents_type_add_form_submit`. (VanDyk 2008, 225)

#### 4.3.2 Lomakkeiden validointi

Lomakkeille voidaan toteuttaa omat validointifunktionsa. Näissä voidaan varmistaa, että kaikki vaaditut kentät on täytetty, ja että ne sisältävät vaaditun tyyppistä tietoa. Lomakkeen määrittelyssä pakollisiksi osoitetut elementit tarkastetaan automaattisesti ilman erillistä validointifunktiota.

## 5 APUVÄLINEITÄ KEHITYKSEEN

Drupalin moduulikehitykseen löytyy yhteisöstä muutamia apuvälineitä. Tässä luvussa keskitytään näistä kahteen, Coder- ja Devel-moduuleihin. Näillä voidaan tarkastella moduulikoodia ja helpottaa vianetsintää.

### 5.1 Coder-moduuli

Coder on laajennusmoduuli, jolla on mahdollista tarkastella, vastaako halutun moduulin koodi Drupalin koodausstandardeja. Lisäksi sillä voidaan muokata moduulin lähdekoodia. Tämän opinnäytetyön toteutusosiossa käytettiin kuitenkin ainoastaan ensimmäisenä mainittua toimintoa.

Ensimmäisen kerran Coder-moduulin koodintarkastus suoritettiin 28.10., kun moduulin lähdekoodi oli toiminnallisuksiensa puolesta valmis. Seuraavat seikat tarkastettiin lähdekoodista:

- **Drupal Coding Standards**  
Tarkastaa, ovatko esimerkiksi muuttujien ja funktioiden nimet oikeanlaisia. Lisäksi tässä käydään läpi mm. sisennykset, tyyppimuunnokset sekä puuttuvat puolipisteet.
- **Drupal Commenting Standards**  
Käy läpi ohjelmakoodin kommentoinnin.
- **Drupal SQL Standards**  
Tarkastaa SQL-lauseiden oikean muodon.
- **Drupal Security Checks**  
Tarkastaa, että käyttäjän syötteet sekä kerätään että tulostetaan turvallisessa muodossa.

Testit suoritettiin tasolla normal, joka jättää mm. sisennykset, turhat rivinvaihdot ja ehtolauseiden muotoiluvirheet huomioimatta. Tällä tasolla testattuna löytyi 24 normaalitason varoitusta. Kolmea varoitusta lukuunottamatta nämä varoitukset sijaitsivat .module-tiedostossa ja koskivat valikon tekstien kääntämistä. (Menu item title and description callbacks, localization 2010)

Testi ajettiin uudestaan tasolla minor, joka ottaa pienimmätkin virheet huomioon. Tuloksena oli edellisten 24 normaalin varoituksen lisäksi 278 pienempää varoitusta. Kaikki nämä virheet korjattiin seuraavin toimenpitein:

.module-tiedostosta poistettiin valikkorakenteen otsikoiden käännökset sekä PHP:n lopetustagi tiedoston lopusta. Kaikkien funktiotiedostojen alkuun lisättiin CVS-avainsana \$Id\$ sekä kommenttiblokki @file. Tiedostoista poistettiin turhat rivinvaihdot sanakäännöksistä alun ja lopun välilyönnit (Check string in t() 2010). Lisäksi camelCase-tyyliset muuttujanimet korjattiin pienellä kirjoitetuksi käyttäen sanaerottimena alaviivaa. Ehtolauseiden rakenteeseen tehtiin korjaukset, joissa else-ehto alkaa omalta riviltään. Lopuksi kaikki PHP:n vakiot korjattiin isolla kirjoitetuksi, esimerkiksi true muutettiin muotoon TRUE.

## 5.2 Devel-moduuli

Drupalin Devel-moduuli tarjoaa apuvälineitä Drupal-kehitystä varten. Se tuo Drupaliin apufunktioita, joilla voidaan tulostaa muuttujia ja taulukoita selkeässä muodossa, esimerkiksi vian etsintää varten. Devel-moduulilla voidaan myös luoda testikäyttöä varten esimerkkisisältöä Drupal-asennukseen. Esimerkkisisältö voi olla muu muassa käyttäjiä, kommentteja tai artikkeleita. Nämä sisällöt saadaan myös poistettua tämän moduulin avulla. Lisäksi tämä moduuli kirjaa ylös suoritusaikeja eri toiminnoille, jolloin voidaan helpottaa sivustolla esiintyvien mahdollisten pullonkaulojen etsimistä ja korjata koodi nopeammaksi. (Drupal Devel 2010)

## 6 XEVENTS-MODUULISSA KÄYTETYT MENETELMÄT

Tässä luvussa kerrotaan opinnäytetyön käytännön toteutusosassa käytetyistä menetelmistä. Näihin kuuluvat muun muassa käytetyt ohjelmat ja ohjelmointikäytännöt sekä lyhyt esittely lopputuotteesta.

### 6.1 Versiohallinta

Työn varsinaisten kooditiedostojen hallintaan otettiin käyttöön Subversion-versiohallinta, joka oli lähinnä ylimääräisenä tallennuspaikkana opinnäytetyön toteutusosan lopputuotteen koodille ja raportin esiversioille. Vertailun (<http://www.svnhostingcomparison.com/>) perusteella otettiin käyttöön Assemblaniminen (<http://www.assembla.com/>) palveluntarjoaja. Se tarjoaa ilmaiseksi muun muassa kahden gigatavun tallennustilan ja WWW-käyttöliittymän.

### 6.2 Tietokanta

Tietokannan suunnittelussa oli alunperin tarkoitus käyttää DBDesigner 4 -ohjelmaa, mutta lopulta päädyttiin käyttämään MySQL Workbenchiä. Tällä ohjelmalla saa suunniteltua tietokannan ja esitettyä sen rakenteen graafisesti. Ohjelmasta on lisäksi saatavana generoitu koodi tietokannan luomiseen, mutta tätä ominaisuutta ei toteutuksessa käytetty, sillä tietokannan ylläpito hoidetaan Drupalin tietokanta-rajapinnalla.

xEventsissä luotiin seitsemän tietokantataulua sekä käytettiin Drupalin olemassa olevista tauluista neljää. *Xevents\_event*-taulu pitää sisällään tiedot luoduista tapahtumista. Tapahtuman tietoja ovat sen nimi ja kuvaus, tapahtuman luoja, sijainti, osallistujien maksimimäärä, hinta vuorokaudelta sekä alku- ja loppupäivämäärät. *Xevents\_type*-taulussa määritellään erilaisten tapahtumatyyppien nimet. *Xevents\_location*-taulu sisältää tapahtumien sijaintipaikat sekä niiden karttaosoitteet. *Xevents\_comment*-taulussa yhdistetään tapahtuma ja kommentti. Kommentin tietoja ovat kommentoija, kommentoitava tapahtuma, itse komment-

ti sekä kommentoinnin ajankohta. *Xevents\_link*-tauluun tallennetaan tapahtumaan liitettäviä hyperlinkkejä. *Xevents\_entry*-tauluun kirjataan tapahtumiin osallistuvat käyttäjät. Taulu sisältää osallistuvan käyttäjän ID-numeron, tapahtuman ID-numeron sekä tiedon siitä, onko kyseinen ilmoittautuminen maksettu. *Xevents\_moderator*-taulussa voidaan määritellä tietyille tapahtumalle ylläpitäjiä yhdistämällä tapahtuman ID-numero halutun käyttäjän ID-numeroon. Tarkempi tietokantakuvaus löytyy liitteestä 1.

Drupalin omista tauluista tärkein on taulu *users*. Se sisältää järjestelmään kirjautumiseen vaaditut tiedot, kuten käyttäjänimen ja salasanan. Käyttäjien roolit määritellään taulussa *role*, joka sisältää järjestelmässä käytettävät roolit. Varsinainen roolien jako tehdään taulussa *users\_roles*, jossa käyttäjille liitetään roolit. Se, mitä milläkin roolilla on oikeus tehdä, määritellään taulussa *role\_permission*. Taulussa liitetään roolille jokin tietty oikeus. Oikeudet tulevat suoraan Drupalin moduuleista

Tietokannan kenttien nimeämisessä päädyttiin käyttämään pieniä kirjaimia ja alaviivaa erottamaan sanat. Drupalin yhteisösivuilta löytyvässä SQL-ohjeistuksessa (<http://drupal.org/node/559302>) ei tästä ollut mainintaa, mutta muissa järjestelmän tietokantatauluissa nimeäminen on toteutettu näin.

### 6.3 Lähdekoodi

xEvents-moduulin varsinainen ohjelmointityö tehtiin Notepad2-nimisellä (<http://www.flos-freeware.ch/notepad2.html>) tekstieditorilla. Se on kevyt ja ilmainen editori, joka osaa värittää eri ohjelmointikielien syntaksia.

Kaikki moduulin varsinaiset toiminnot luokkien funktioita lukuun ottamatta ovat moduulihakemiston juuressa omissa tiedostoissaan. Koodi on jaettu omiin tiedostoihinsa riippuen siitä, mihin moduulin osaan toiminto liittyy. Esimerkiksi tiedostossa *xevents.events.inc.php* on kaikki funktiot, jotka liittyvät varsinaisten tapahtumien käsittelyyn. Muita tiedostoja ovat esimerkiksi *xevents.users.inc.php* ja *xevents.locations.inc.php*.

### 6.3.1 Luokat

Moduulin ohjelmoinnissa käytettiin hyväksi PHP:n olioita ja periytymistä. Luokkien välisten suhteiden hahmottamiseen käytettiin ilmaista StarUML-ohjelmaa (<http://staruml.sourceforge.net/en/>). Luokkakaavio on nähtävissä liitteessä 2.

Luokkien funktioiden nimeämisessä käytettiin "lowerCamelCase"-käytäntöä (Naming conventions 2010), jossa luokan funktiot nimetään siten, että nimi alkaa pienellä kirjaimella ja jokainen seuraavista sanoista isolla kirjaimella, esimerkiksi `getDbTable()`. Muuttujat kirjoitettiin pienillä kirjaimilla ja välilyönnit korvattiin alaviivalla. Näin saatiin sekä luokkiin että tietokantaan yhdenmukaiset nimet.

Yliluokkana toimii luokka `XEventsAbstract.php`. Tämä sisältää kaikille muille luokille pakotetut muuttujat. Luokalla on ilmentymämuuttujat `db_table` ja `primary_key`. Muuttuja `db_table` pitää sisällään tiedon kyseisen luokan tietokantataulusta, jonne luokasta muodostettujen olioiden tiedot tallennetaan. Muuttuja `primary_key` taas kertoo kyseisen luokan tietokantataulun pääavaimen, jolla tiedot tietokannassa yksilöidään.

Lisäksi yliluokka sisältää kaikille periville luokille yhteisiä funktioita tiedon käsittelyyn. `__construct` on oletusmuodostin. Muodostimessa hyödynnetään `db_table`-muuttujaa, jolloin jokaiselle muodostetulle oliolle määritetään ominaisuudet suoraan tietokannan taulun kenttiä vastaaviksi. `Add`-funktio tallentaa olion tiedot tietokantaan `db_table`-muuttujan määrittelemään tauluun. `Load`-funktio lataa olion tiedot suoraan olion muuttujiksi tietokannasta `db_table`-muuttujan määrittelemästä taulusta. `Save` päivittää tietokantaan olion tiedot ja `Delete` poistaa ne.

### 6.3.2 Funktiotiedostot

Moduulin eri toiminnot sijaitsevat funktiotiedostoissa (`include/*inc.php`). Funktioiden ja muuttujien nimet merkittiin pienellä ja sanojen erottimena käytettiin alaviivaa. Eri moduulin toiminnot jaettiin eri tiedostoihin selkeyden vuoksi.

### 6.3.3 Hakemistorakenne

Puhtaassa Drupal-asennuksessa ydinmoduulit sijaitsevat alihakemistossa *modules*. Kolmansien osapuolten tuottamat lisäosat taas asennetaan alihakemistoon nimeltä *sites*. Tämä hakemisto sisältää oletuksena hakemiston *all*. Moduulit tallennetaan tämän hakemiston sisältä löytyvän *modules*-hakemiston sisään.

Kun moduuli asennetaan *all*-hakemiston alle, on se käytettävissä jokaisella tämän Drupal-asennuksen sivustoista. Mikäli moduuli halutaan rajata vain tietyn sivuston käyttöön, tehdään *sites*-hakemistoon tälle sivustolle oma alihakemisto, johon halutut moduulit asennetaan. Tällöin myös *sites.php*-konfigurointitiedostoa pitää muokata vastaavasti. Mutta koska tämän opinnäytetyön toteutusosa on tehty ainoana sivustona, ei tämä raportti käsittele ns. multi site -asennusta tämän enempää.

Hakemiston *sites/all/modules* sisään siirretään kansio, joka sisältää moduulin tarvitsemat tiedostot.

Moduulikansion juuressa sijaitsevat moduulin *.info*-, *.module*- ja *.install*-tiedostot. Näiden tiedostojen merkitys on selvitetty aiemmin raportissa. Alihakemistossa *include* sijaitsevat varsinaiset moduulin funktiotiedostot, jotka vastaavat esimerkiksi lomakkeiden rakenteesta sekä niiden käsittelystä. Alihakemisto *class* sisältää moduulin käyttämät PHP-luokat. Alihakemistossa *res* sijaitsee erilaiset resurssit, joita moduuli tarvitsee. Näitä ovat JavaScript-tiedostot ja CSS-tyylitiedostot.

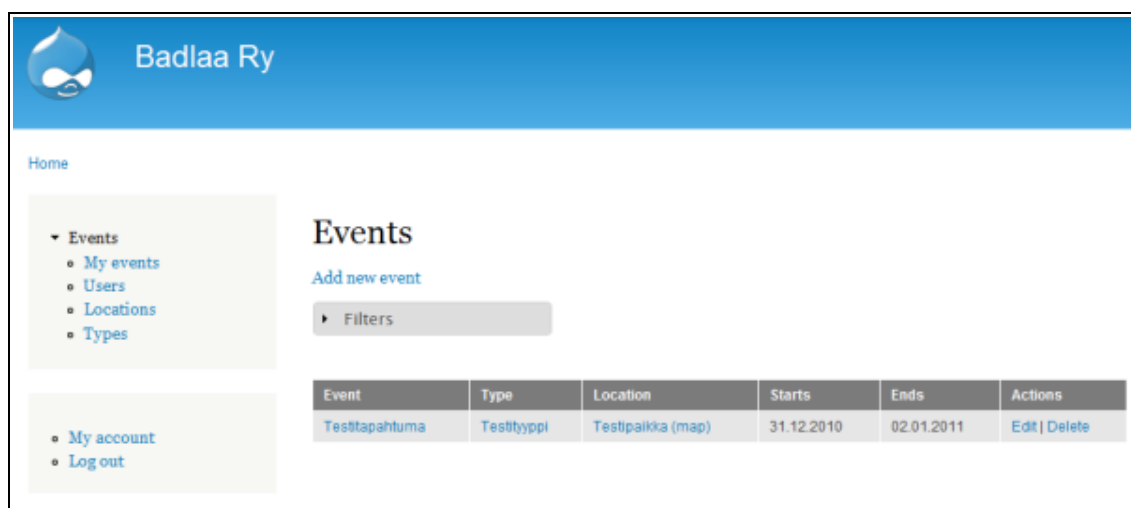
## 6.4 Testaus

Varsinaisia ohjelmistotestauksen keinoja ei opinnäytetyön toteutusosan loppu-tuotteen testauksessa käytetty. Aluksi kaikki moduulin lomakkeet testattiin siten, että kenttiin yritettiin syöttää arvoja, jotka eivät niihin kuuluneet. Näin koodissa havaitut bugit korjattiin. Tätä toistettiin kunnes lomakkeista ei enää löytynyt virheitä.

Seuraava vaihe oli syöttää järjestelmään oikeanlaista testidataa ja tarkastaa, että kaikki tallentuu normaalisti. Kun tietojen tallennus oli tarkastettu, käytiin niiden tulostusnäkyvät läpi. Tarkastettuja asioita oli muun muassa päivämäärän oikeanlainen muotoilu ja lukujen oikeat pyöristystarkkuudet. Lopuksi testattiin vielä, että linkit ja erityisesti Takaisin-linkit vievät oikeaan paikkaan.

## 6.5 xEventsin käyttö

Järjestelmässä on kahdentasoisia käyttäjiä: normaalikäyttäjiä ja ylläpitäjiä. Ylläpitäjän roolilla navigointivalikossa näkyy laajempi valikoima linkkejä. Kuvassa 7 näkyy ylläpitäjän etusivu.



Kuva 7: Ylläpitäjän etusivu

Normaalin käyttäjän navigaatiovalikosta puuttuvat linkit *Users*, *Locations* ja *Types*. Näiden kautta voidaan hallinnoida tapahtumien sijaintipaikkoja sekä tapahtumatyyppejä. Lisäksi normaalikäyttäjältä puuttuvat linkit *Edit* ja *Delete* tapahtumarivin lopusta. Painamalla *Filters*-painiketta avautuu valikko, josta tapahtumia voidaan suodattaa tyyppin, sijaintipaikan ja ajankohdan mukaan.

*Types*-linkkiä klikkaamalla avautuu listaus järjestelmään luoduista tapahtumatyypeistä. Kuvassa 8 näkyy tyyppilistaus.



Type	Actions
Testityyppi	Edit   Delete

Kuva 8: Tapahtumatyyppien listaus

Tyydin nimeä klikkaamalla avautuu yhteenvetosivu, jossa listataan kaikki tämän tyydin tapahtumat. Ylläpitäjille on näkyvissä linkit tapahtumatyydin muokkaukseen (Edit) ja poistamiseen (Delete). Näiden lisäksi sivun otsikon alapuolella on esillä linkki, jonka kautta voidaan järjestelmään syöttää uusia tapahtumatyyppiä. Kuvassa 9 näkyvät kaikki tapahtumat, joiden tyyppi on Testityyppi.



Event	Location	Starts	Ends
Testitapahtuma	Testipaikka (map)	31.12.2010	02.01.2011

Kuva 9: Kaikki Testityyppi-tapahtumat

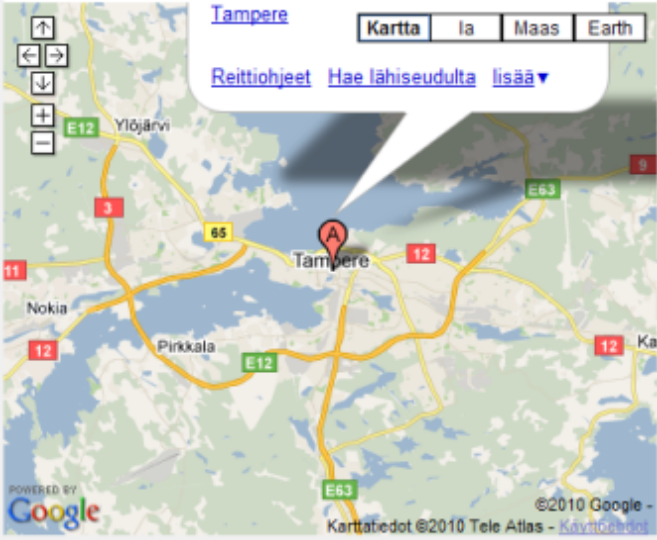
Tapahtuman nimeä klikkaamalla avautuu yhteenvetosivu, jossa näytetään tapahtuman tiedot. Tapahtumapaikan nimestä puolestaan avautuu tapahtumapaikan yhteenvetosivu, jossa näytetään tapahtumapaikan tiedot. Näitä tietoja ovat tapahtuman nimi, kartta tapahtumapaikalle sekä kaikki tämän tapahtumapaikan tapahtumat. Mikäli käyttäjällä on ylläpitäjän rooli järjestelmässä, ovat linkit *Edit* ja *Delete* näkyvissä heti sivun otsikon alapuolella. Näiden linkkien kautta päästään muokkaamaan tapahtumatyyppiä tai poistamaan se ja siihen luodut tapahtumat. Kuvassa 10 näkyy tapahtumapaikan Testipaikka yhteenvetosivu.

## Location Testipaikka

[Edit](#) | [Delete](#)

Location name: Testipaikka

Location map:



©2010 Google - Karttatiedot ©2010 Tele Atlas - Käyttöoikeudet

[Näytä suurempi kartta](#)

Event	Type	Starts	Ends
<a href="#">Testitapahtuma</a>	<a href="#">Testityyppi</a>	31.12.2010	02.01.2011

Kuva 10: Tapahtumapaikan yhteenvetosivu

Navigaatiolinkkiä *Users* klikkaamalla avautuu käyttäjälistaus, jossa näkyvät kaikki järjestelmään luodut käyttäjät. Kuvassa 11 näkyy käyttäjälistaus.

## Users

Users
<a href="#">Aaltonen, Antti</a>
<a href="#">Esimerkki, Erkki</a>
<a href="#">Heinäjärvi, Hannu-Pekka</a>
<a href="#">Hovi, Seppo</a>
<a href="#">Jutila, Timo</a>
<a href="#">Laitinen, Petri</a>
<a href="#">Salminen, Simo</a>

Kuva 11: Käyttäjälistaus

Käyttäjän nimeä klikkaamalla päästään käyttäjän yhteenvetosivulle. Sivulla näytetään käyttäjän perustiedot sekä kaikki tapahtumat, joihin käyttäjä on ilmoittau-

tunut. Kuvassa 12 näkyy käyttäjän yhteenvetosivu.

User Hannu-Pekka Heinäjärvi					
Fullname:	Heinäjärvi Hannu-Pekka				
Username:	hannupekka				
Mail:					
Event	Type	Location	Starts	Ends	Is paid
Testitapahtuma	Testityyppi	Testipaikka	31.12.2010	02.01.2011	No

*Kuva 12: Käyttäjän yhteenvetosivu*

Kuvassa 12 näkyvä yhteenveto on saatavilla myös navigaatiolinkin *My events* kautta. Tällöin sivulla näytetään sisäänkirjautuneen käyttäjän tiedot ja tapahtumat. Ylläpitäjän roolilla etusivulla on näkyvillä linkki *Add new event*, jonka kautta voidaan syöttää järjestelmään uusia tapahtumia.

Tapahtuman yhteenvetosivulta voidaan normaalikäyttäjän roolilla nähdä tapahtuman perustiedot, oman ilmoittautumisen tila, hintatiedot, mukaan ilmoittautuneet käyttäjät sekä kommentit ja linkit. Kommentteja ja linkkejä voidaan tätä kautta myös lisätä. Kuvassa 13 näkyy tapahtuman kommenttillistaus ja lomake uuden kommentin lisäämiseksi.

▼ Comments

**Comment:**

Add comment

---

**Comments**

Testikommentti  
Erkki Esimerkki @ 15.12.2010 19:32:57

*Kuva 13: Tapahtuman kommenttillistaus ja uuden kommentin lisäys*

Lomake uuden kommentin lisäämiseksi on kommenttillistauksen alussa. Sen jälkeen on listattuna kaikki tapahtumaan lisätyt kommentit alkaen vanhimmasta.

Kuvassa 14 näkyvät tapahtuman perustiedot ja oman ilmoittautumisen tila.

## Testitapahtuma

*Tapahtuman testausta*

**Details**

Event name:	Testitapahtuma
Type:	<a href="#">Testityyppi</a>
Location:	<a href="#">Testipaikka (map)</a>
Organizer:	Hannu-Pekka Heinäjärvi
Entrants:	Unlimited
Price per day:	Free
Date:	31.12.2010 - 02.01.2011, 3 days

You **are** enrolled in this event.

[Leave event](#)

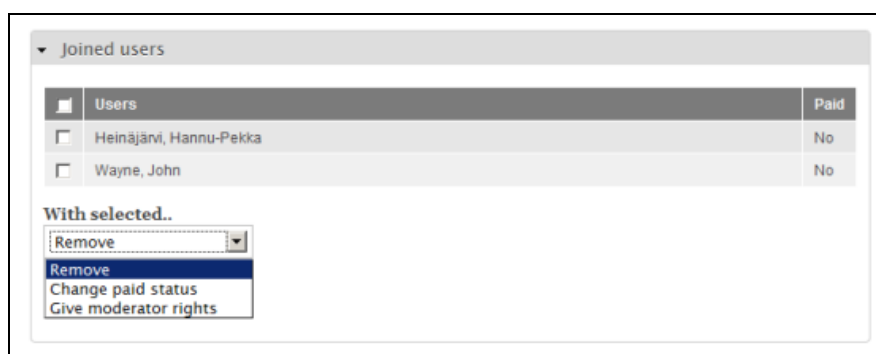
**Pricing**

Total price:	Free
Price per user:	Free

*Kuva 14: Tapahtuman perustiedot*

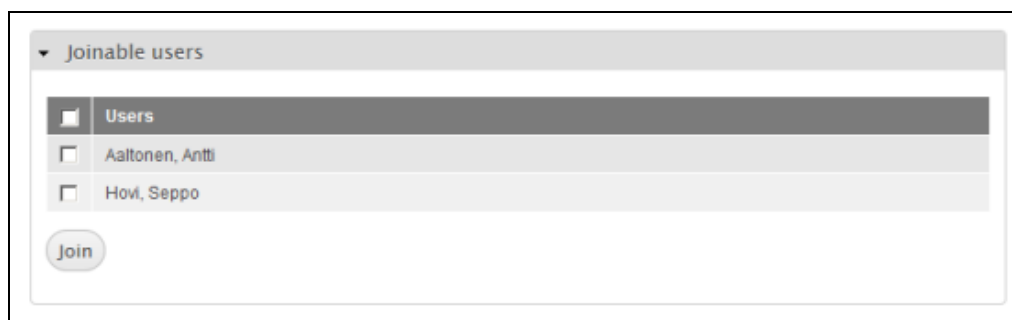
Kuvan 14 keskivaiheilla näkyy oman ilmoittautumisen tila. Oma ilmoittautuminen voidaan perua klikkaamalla linkkiä *Leave event*. Mikäli tapahtuma on määritelty kaikille avoimeksi, voidaan tältä sivulta myös ilmoittautua tapahtumaan. Tällöin linkki *Leave event* on korvattu linkillä *Join event*. Ylläpitäjän käyttäjäroolilla tällä sivulla voidaan hallinnoida tapahtuman osallistujia, kommentteja, linkkejä sekä tapahtuman ylläpitäjiä. Lisäksi ylläpitäjän roolilla on näkyvissä linkit *Edit* ja *Delete*, joiden kautta tapahtumaa voi muokata tai sen voi poistaa.

Kohdassa *Pricing* ilmoitetaan tapahtuman kokonaishinta ja hinta jokaiselta käyttäjältä. Kuvan 14 esimerkissä tapahtuma on ilmainen, joten hintatietoja ei näy. Kohdassa *Total price* näytetään tapahtuman kokonaishinta ja sen alapuolella, kohdassa *Price per user*, näytetään kokonaishinta jaettuna osallistujien lukumäärällä. Kuvassa 15 on esitettyä osallistujien hallintaa.



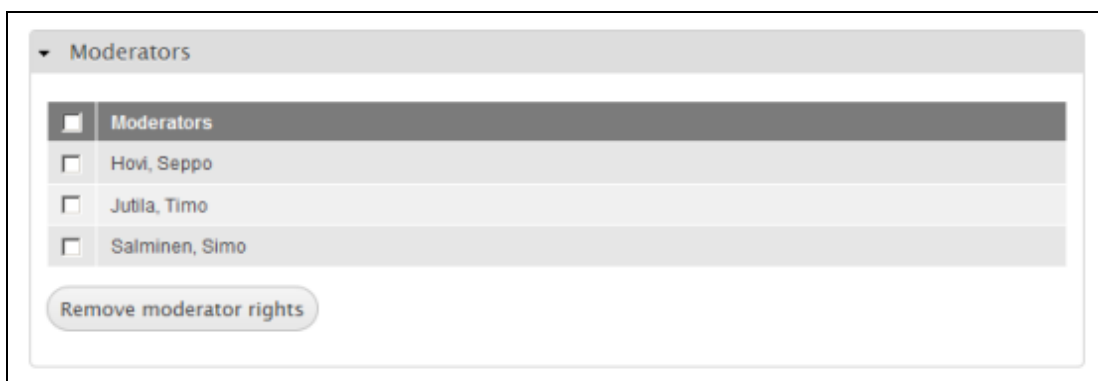
Kuva 15: Tapahtuman osallistujien hallinta

Kuvassa 15 näkyvästä käyttäjälistasta valitaan halutut käyttäjät ja alasvetovalikosta *With selected* voidaan valita näille käyttäjille tehtävä toimenpide. *Remove* poistaa käyttäjän tapahtumasta, *Change paid status* merkkää käyttäjän maksaneeksi tai poistaa tiedon maksusta ja *Give moderator rights* lisää käyttäjän tämän tapahtuman ylläpitäjäksi. Tällöin käyttäjä saa tapahtumaan täydet ylläpitäjän oikeudet. Ylläpitäjät voivat tapahtumasivulta lisätä tapahtumaan osallistujia. Tätä ominaisuutta tarvitaan silloin, kun tapahtuma on asetettu suljetuksi kaikilta. Kuvassa 16 näkyy käyttäjälistaus, josta käyttäjiä voidaan lisätä tapahtumaan.



Kuva 16: Käyttäjien lisäys tapahtumaan

Käyttäjälistasta valitaan halutut käyttäjät ja painiketta *Join* painamalla valitut käyttäjät lisätään tapahtuman osallistujiin. Tapahtuman ylläpitäjiä voidaan hallinnoida kohdassa *Moderators*. Tämä näkymä on esiteltyä kuvassa 17.



*Kuva 17: Tapahtuman ylläpitäjien hallinta*

## 7 JOHTOPÄÄTÖKSET JA POHDINTA

Opinnäytetyön käytännön toteutusosan, xEvents-moduulin, tavoitteeksi oli asetettu henkilökohtainen kehittyminen ja asiakkaan tarpeiden tyydyttäminen. Moduulin suunnitteluun sain melko vapaat kädet asiakkaalta. Ohjelmoinnissa en juurikaan törmännyt suurempiin ongelmiin, kiitos huolellisen suunnittelun. Apuvälineinä käytetyt Star UML ja MySQL Workbench auttoivat paljon kokonaisuuden hahmottamisessa. Vaikeinta opinnäytetyössä oli tämän raportin jäsentely ja kirjoittaminen.

Kokonaisuutena asiakkaan näkökulmasta arvioisin suoriutuneeni vähintäänkin tyydyttävästi. Käytännön toteutusosassa suunnitellun ja ohjelmoidun moduulin toiminnallisuus vastaa hyvin pitkälti alkuperäisiä suunnitelmia. Testauksen aikana ei myöskään ilmennyt ongelmia lopputuotteen toiminnassa.

Henkilökohtaisen kehittymisen taso Drupal-ohjelmoinnissa ei aivan vastannut omia odotuksiani. Opin toki uutta sekä Drupaliin että PHP-ohjelmointiin yleisesti liittyen, mutta etenkin Drupalin AJAX-ominaisuuksiin olisi voinut keskittyä enemmänkin. Suunnittelussa ja määrittelyssä koen edistyneeni melko paljon. Koska asetin käytännön toteutusosalle löysähkön aikataulun, saatoin käyttää normaalia enemmän aikaa koodirakenteiden ja tietokannan suunnitteluun. Tästä johtuen itse ohjelmointityö eteni sujuvasti.

Tämän raportin tavoitteeksi asetin Drupal-kehityksen perusteiden selvittämisen sekä käytännön xEvents-moduulin esittelyn lukijalle. Mielestäni perusteet Drupalista ja kehityksestä Drupal-alustalla esiteltiin keskinkertaisesti. Drupal-järjestelmä on sen verran laaja, että tarkempi rajausta olisi ehkä ollut tarpeen. Myöskin lopputuotteen perusteellinen esittely osoittautui hankalaksi, sillä kunnan ruutu-kaappauksia on vaikea tähän raporttiin upottaa. Tämä johtuu siitä, että sovelluksen sivuilla on paikoittain paljon tietoa, joka taas vaatii suurikokoisia kuvia. Niitä ei kuitenkaan raporttiin saa sopivasti aseteltua. Paras perehdytys lopputuotteen olisi ehdottomasti esittely suoraan selaimessa.

Jatkokehityksenä voisi tuotetun Drupal-moduulin valikkorakennetta yksinkertaistaa kooditasolla. Nykyisellään valikossa käytetään turhan montaa valikko-objektia, kun sama toiminnallisuus olisi saavutettavissa vähäisemmälläkin määrällä. Lisäksi moduulin lomakkeita voisi mahdollisesti käyttää monipuolisemmin. Tässä versiossa moduulista jäivät vielä suomenkieliset käännökset toteuttamatta, vaikka lokalisointi onkin koodissa otettu huomioon.

## LÄHTEET

Check string in t(). 2010. [www-sivu]. [Luettu 28.10.2010]. Saatavissa: <http://drupal.org/node/304150>.

Converting 6.x modules to 7.x. 2010. [www-sivu]. [Luettu 11.10.2010]. Saatavissa: <http://drupal.org/node/224333#install-schema>.

Database functions: db\_add\_field. 2010. [www-sivu]. [Luettu 11.10.2010]. Saatavissa: [http://api.drupal.org/api/drupal/includes--database--database.inc/function/db\\_add\\_field/7](http://api.drupal.org/api/drupal/includes--database--database.inc/function/db_add_field/7).

Database general concepts. 2010. [www-sivu]. [Luettu 25.11.2010]. Saatavissa: <http://drupal.org/node/310070>.

Delete queries. 2010. [www-sivu]. [Luettu 25.11.2010]. Saatavissa: <http://drupal.org/node/310081>.

Drupal 7 Release Date. 2010. [www-sivu]. [Luettu 16.9.2010]. Saatavissa: <http://drupal7releasedate.com/>.

Drupal Devel. 2010. [www-sivu]. [Luettu 28.10.2010]. Saatavissa: <http://drupal.org/project/devel>.

Drupal's code registry. 2010. [www-sivu]. [Luettu 8.10.2010]. Saatavissa: <http://drupal.org/node/350780>.

Dynamic queries. 2010. [www-sivu]. [Luettu 25.11.2010]. Saatavissa: <http://drupal.org/node/310075>.

Form API reference. 2010. [www-sivu]. [Luettu 25.11.2010]. Saatavissa: [http://api.drupal.org/api/drupal/developer--topics--forms\\_api\\_reference.html/7](http://api.drupal.org/api/drupal/developer--topics--forms_api_reference.html/7).

Insert queries. 2010. [www-sivu]. [Luettu 25.11.2010]. Saatavissa: <http://drupal.org/node/310079>.

Menu item title and description callbacks, localization. 2010. [www-sivu]. [Luettu 28.10.2010]. Saatavissa: <http://drupal.org/node/140311>.

Naming conventions. 2010. [www-sivu]. [Luettu 16.9.2010]. Saatavissa: <http://drupal.org/coding-standards#naming>.

Run multiple sites from the same code base. 2010. [www-sivu]. [Luettu 8.10.2010]. Saatavissa: <http://drupal.org/getting-started/6/install/multi-site>.

Schema API. 2010. [www-sivu]. [Luettu 11.10.2010]. Saatavissa: <http://drupal.org/developing/api/schema>.

Schema API reference. 2010. [www-sivu]. [Luettu 11.10.2010]. Saatavissa: <http://drupal.org/node/146939>.

System requirements. 2010. [www-sivu]. [Luettu 7.10.2010]. Saatavissa: <http://drupal.org/requirements>.

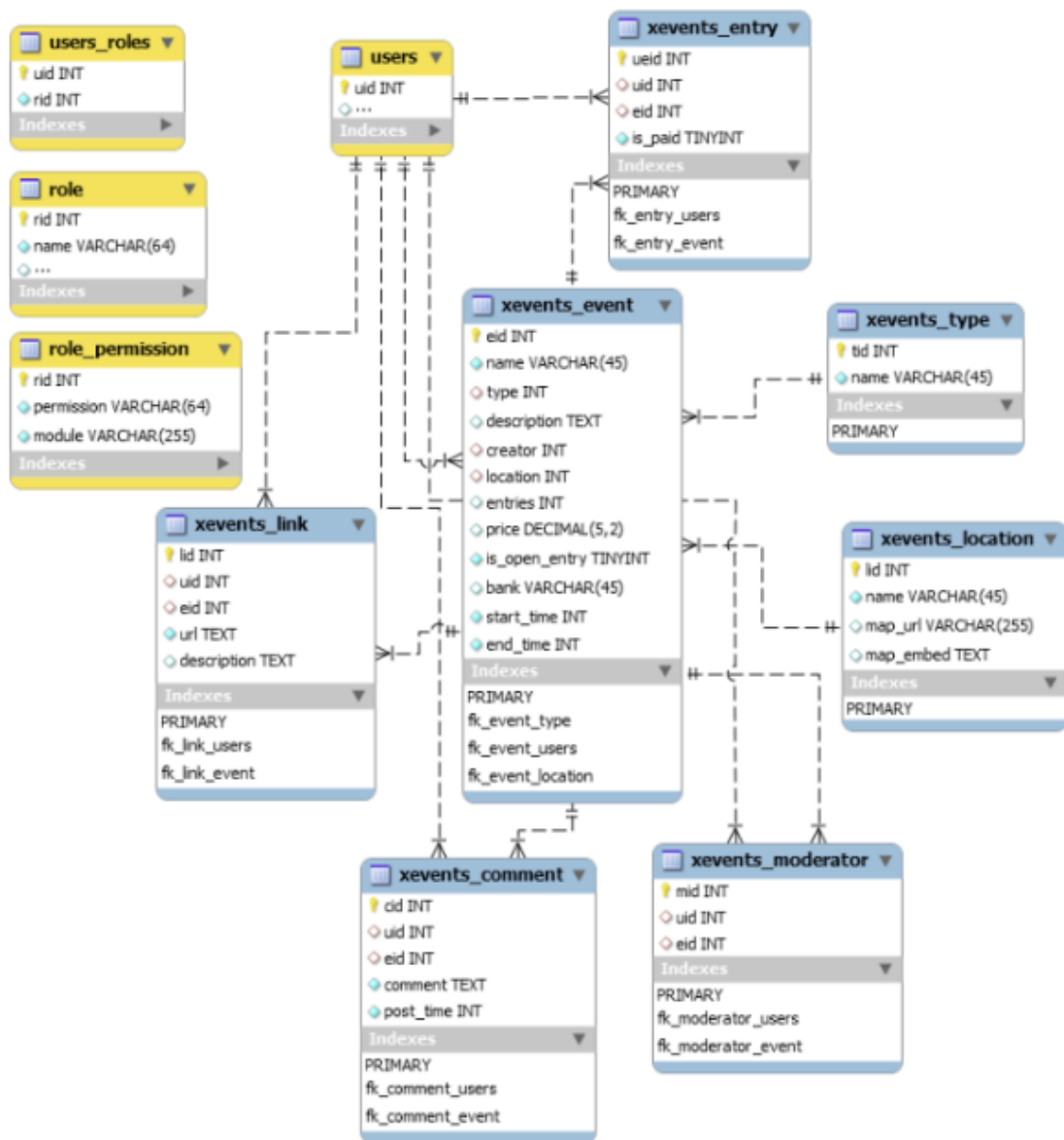
Update queries. 2010. [www-sivu]. [Luettu 25.11.2010]. Saatavissa: <http://drupal.org/node/310080>.

Using t-function. 2010. [www-sivu]. [Luettu 19.11.2010]. Saatavissa: <http://api.drupal.org/api/drupal/includes-bootstrap.inc/function/t/7>.

VanDyk, J. 2008. Pro Drupal Development. Berkeley, CA: Apress.

Writing .info files. 2010. [www-sivu]. [Luettu 16.9.2010]. Saatavissa: <http://drupal.org/node/542202>.

LIITE 1  
TIETOKANTAKAAVIO



LIITE 2

LUOKKAKAAVIO

