

Tallinna Ülikool

Informaatika Instituut

# ÄRIETTEVÕTTE VEEBILEHE LOOMINE JOOMLA BAASIL

Seminaritöö

Autor: Hannes Liiker

Juhendaja: Jaagup Kippar

Autor: ..... , 2011

Juhendaja: ..... , 2011

# Sisukord

SISUKORD .....	1
SISSEJUHATUS.....	2
1. JOOMLA .....	3
1.1 ARHITEKTUUR.....	3
1.2 VÕIMALUSED.....	4
1.3 LAIENDUSED .....	5
2. JOOMLA DISAINIMALL .....	6
2.1 LISAMINE.....	6
2.2 HALDAMINE.....	6
2.3 KATALOOGI STRUKTUUR .....	7
2.4 ANATOOMIA .....	9
2.4.1 INDEX.PHP.....	9
2.4.2 TEMPLATEDetails.XML .....	14
2.5 JDOC LAUSED.....	15
2.6 PARAMEETRID .....	17
2.7 MOODULI POSITSIOONI STIILID .....	19
2.8 HTML ÜMBERKIRJUTISED .....	19
3. OÜ ANTIIGIHUNT DISAINIMALLI LOOMINE .....	20
3.1 KRITERIUMID .....	20
3.2 PLANEERIMINE .....	21
3.3 GRAAFILISE KAVANDI LOOMINE .....	21
3.4 KODEERIMINE.....	21
3.5 SIDUMINE JOOMLAGA.....	22
3.6 TESTIMINE .....	24
KOKKUVÕTE.....	25
KASUTATUD KIRJANDUS .....	26

# Sissejuhatus

Mitte kunagi varem pole tarkvara kasutajaliidesele pööratud niipalju tähelepanu, kui praegu seoses veebirakendustega. Internet on esmaseid kohti informatsiooni otsimiseks. Veebilehe väljanägemine ja kasutajamugavus on peamised faktorid, mis loovad esmamulje, mille põhjal tihtilugu tehakse ka reaalsed otsused.

Joomla sisuhaldusüsteem on tuntud oma laienduste rohkuse poolest, kuid enamus neist on tasuta, mis kehtib ka disainimallide puhul. Soetades tasuta disainimalli ei saa ka garantiid, et kusagil interneti sügavustes pole veebilehte, mis on 99% sama välimusega.

Käesoleva seminaritöö eesmärgiks on välja töötada äriettevõtte veebilehe disainimall Joomla sisuhaldussüsteemile. Eesmärgi saavutamise peaks andma ülevaate, kui paindlik on Joomla süsteem rakenduse kasutajaliidese loomisel.

Lahenduse piiritlemiseks mõtles autor välja fiktiivse kodulehe tellija, kelleks on ettevõtte nimega OÜ Antiigihunt. Töö on koostatud tuginedes Joomla versioonile 1.5.20.

Antud uurimustöö peaks tooma vastuse järgmisele küsimusele:

Kas kõik graafilised kavandid, mis on realiseeritud HTML keeles, on võimalik siduda Joomla süsteemiga, tegemata seejuures suuremaid mööndusi?

Tulemuste saavutamiseks on autor seadnud endale järgmised ülesanded:

- Tutvuda Joomla omaduste, terminite ja võimalustega.
- Tutvuda Joomla disainimalli ülesehituse ja tööpõhimõtetega.
- Luua kriteeriumid rakenduse loomiseks
- Reaalne rakenduse loomine ja evalvatsioon

Esimene peatükk teeb ülevaate Joomla olemusest ja terminitest. Teine peatükk keskendub konkreetsele tahule ehk disainimallile, tehes ülevaate nähtuse struktuurist, tööpõhimõtetest ja võimalustest ja luues seega eeldused viimasele peatükile, kus valmib konkreetne rakendus.

Sooviksin avaldada tänu oma juhendajale Jaagup Kipparile operatiivse abi ja meeldiva koostöö eest.

# 1. Joomla

Joomla sisuhaldussüsteem (*CMS - Content Management System*) on info edastamiseks veebilehtedel ja intranetis. Lihtsamalt öeldes on Joomla tarkvara, mille põhjal on võimalik valmistada veebilehekülgi ja erinevaid veebiaplikatsioone, nt väikesed personaalsed koduleheküljed kuni suuremahuliste korporatiivportaalideni välja. (Harwani, 2009)

Sisuhaldussüsteemi võib piltlikult nimetada kui raamatukoguks, mis arvestab, korrastab ja arhiveerib raamatuid. Raamatute asemel on veebis näitena tekstid, pildid, muusika ja videod. (Joomla!, 2010)

2009. aasta oktoobris avaldatud sisuhalduste turuosa uuring (*2009 Open Source CMS Market Share Report*) näitas, et Joomla on kõige populaarsem avatud koodiga sisuhaldussüsteem maailmas. (CMSWire, 2009)

Joomla on avatud lähtekoodiga ja tasuta kättesaadav – välja antud GNU litsentsi alusel. GNU ehk Üldine Avalik Litsents on litsents vaba tarkvara jaoks, eesmärgiga anda kasutajale õigused kopeerida, modifitseerida ja levitada programme ning tagada, et ka uutel kasutajatel oleksid samad võimalused. Erandlikult võrreldes teiste autoriõiguse seadustega kehtivad need õigused ka äritegevuses. (GNU, 2010)

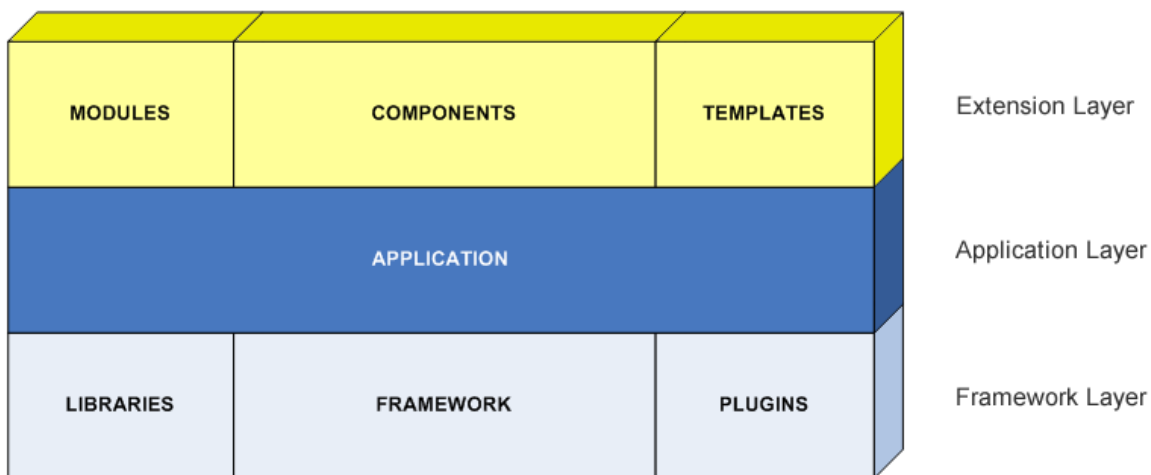
## 1.1. Arhitektuur

Joomla on PHP programmeerimiskeeles kirjutatud rakendus, mis käivitatakse veebiserveris, kus viiakse läbi ka kõik operatsioonid. Veebilehe info salvestamiseks kasutatakse MySQL andmebaase.

Joomla koosneb kolmest kihist: raamistiku, aplikatsiooni ja laienduste kiht. Kihid on vajalikud hoidmaks organiseeritust ja paindlikkust. (Lanham & Kennard, 2010)

1. **Raamistiku kiht** (*Framework layer*) - Sisaldab fundamentaalseid funktsioone, millest tulenevad ja sõltuvad ülejäänud kihid.
2. **Aplikatsiooni kiht** (*Application layer*) – Sisaldab aplikatsioone, mis laiendavad raamistiku kihi funktsioone.

3. **Laienduste kiht** (*Extension Layer*) – Funktsioonide laiendamiseks kasutades mooduleid, komponente ja disainimalle.



Joonis 1. Joomla arhitektuur

## 1.2 Võimalused

Ülevaade Joomla baaspaketiga kaasnevatest võimalustest.

- WYSIWYG (What You See Is What You Get) Content Editor – Tekstitöötlusprogramm
- Content scheduling – Võimalus ajaliselt määrata, millal sisu avaldatakse.
- Content archiving – Artiklite archiveerimine.
- User management – Kasutajate lisamine ja grupeerimine.
- Access Control – Kasutamissoiguste määratlemine.
- Media Manager – Meediafailide organiseerimine.
- Language Manager – Mitmekeelsete lehtede tegemine ja organiseerimine.
- Banner Manager – Reklaami lisamine ja haldamine.
- Contact Manager – Kasutajate andmete salvestamine ja organiseerimine.
- Polls – Küsitluste ja uuringute lisamine.
- Search – Otsingumootori lisamine.
- Web links management – Linkida tegemine veebilehtedele, mis asuvad väljaspool süsteemi.
- Content syndication – Sisu avaldamine RSS voona.
- News feed aggregation and display – Välise RSS voode lisamine.

- Integrated help system – Kasutusjuhend.
- Multiple template management – Uute disainimallide lisamine ja muutmine.
- Cache management – Buffermälu kasutamine.
- Integrated FTP – Sisseehitatud FTP Klient.
- Search engine friendly URLs – Otsingumootoritele kergesti leidavate linkide loomine.

## 1.3 Laiendused

Joomla koosneb viit tüüpi laiendustest. (Sevardia & Crowder, 2010)

- **Komponent** (*component*) – Reeglina kõige mahukam Joomla laiendus. Üldjuhul on komponent eraldiseisev rakendus või lisaprogramm, mis lisab funktsionaalsust (nt e-pood või foorum). Võib koosneda mitmest moodulist ja pluginist. Iga Joomla loodud veebirakendus koosneb vähemalt ühest komponendist, üldjuhtudel on see *com\_content*, mis kuvab sisu.
- **Moodul** (*modul*) – Kasutatakse komponentide lisadena, kuid võib olla ka iseseisev rakendus. Moodulid lisavad funktsionaalsust (nt menüüd või küsitlusvormid).
- **Plugin** (*plugin*) – Kasutatakse protsesside realiseerimiseks (nt otsimine või sisu filtreerimine).
- **Disainimall** (*template*) – Lehekülje välimuse ja struktuuri muutmiseks.
- **Keel** (*language*) – Võimalus lisada erinevaid tõlkeid.

## 2. Joomla disainimall

Teemakujundus, kujundusmudel, template jms – autor kasutab antud Joomla laienduse nimetamiseks sõna *disainimall*.

Disainimall on kasutajaliidese raamistik, mille piires toimub lehekülje sisu ja erinevate komponentide dünaamiline kuvamine. Enamlevinud koostise moodustavad tehnoloogiad nagu HTML, PHP, CSS, XML, JScript ja pildid formaatides PNG, GIF ja JPG.

Veebilehestikul on võimalik kasutada mitut disainimalli ehk igale leheküljele saab määrata oma malli. Ühe disainimalli kasutamine terve veebilehe peale on mugav, kuna muudatusi tehes tuleb käsitleda ainult ühte faili. (Sevardia & Crowder, 2010)

Eristatakse kahte tüüpi disainimalle – *frontend* ja *backend*. Esimene on veebilehe väline väljund, ehk kõik see, mida külastaja näeb sattudes veebilehte. Teine on administratiivliidese kuvamiseks, mis on mõeldud eeskätt veebilehe haldajate töö lihtsustamiseks ja mida tavakülastajad ei näe. (Rhamel, 2009)

Antud töös keskendub autor ainult frontend disainimallidele.

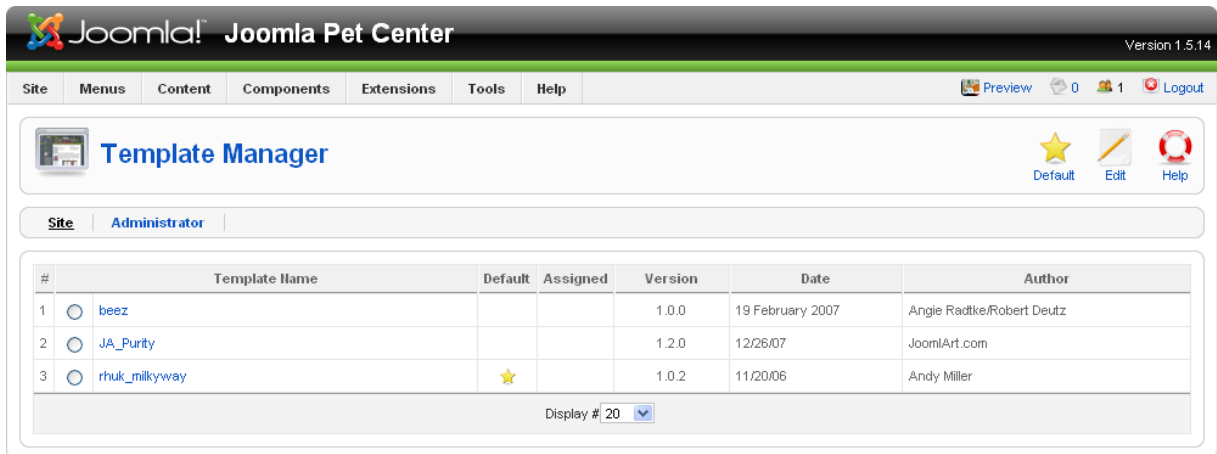
### 2.1 Lisamine

Disainimallide lisamiseks on olemas kaks võimalust: installeerimine läbi laienduste halduri (*Extension Manager*) või manuaalne failide lisamine */templates* kataloogi. (Sevardia & Crowder, 2010)

Automaatsel installeerimisel peavad disainimalli kõik failid olema kokkupakitud kas ZIP või TAR formaati.

### 2.2 Haldamine

Disainimallide aktiveerimiseks ja muutmiseks on Joomla administratiivliides nimega *Template Manager*, mille leiab laienduste menüüst (*Extensions ->Template Manager*). Liides võimaldab reaajas redigeerida *index.php* koodi ja kõiki kaasnevaid CSS stiililehti. (Sevardia & Crowder, 2010)








Joonis 2. Disainimallide haldusliides (Template Manager)

## 2.3 Kataloogi struktuur

Kõik frontend disainimallid asuvad Joomla alamkaustas */templates*. Igal kaustal on oma unikaalne nimi.

Järgnev joonis tutvustab elementaarse disainimalli kataloogi (vt Joonis 3).

Joomla / templates / disainimalli_nimi /		
Name	Type	Size
 css	File folder	
 images	File folder	
 index	PHP Script	2 KB
 template_thumbnail	Adobe Fireworks P...	10 KB
 templateDetails	XML File	2 KB

Joonis 3. Näide lihtsakoelise disainimalli kataloogist

Ülevaade kataloogi sisust:

- **css kaust** - Sisaldab CSS stiililehti.
- **images kaust** – Sisaldab kasutatavaid pilte.



- **index.php** – Disainimalli rakendamisel käivitatakse antud fail esimesena. Defineerib disainimalli ülesehituse ja koostise.
- **templateDetails.xml** – Koosneb meta-andmete nimekirjast.

Kataloogi sisu võib varieeruda mitmeti, olenevalt looja käekirjast ning kui mahuka ja keerulise malliga on tegemist. Minimaalse suurusega disainimall koosneb kahest failist: *index.php* ja *templateDetails.xml*.

Joomla baaspaketiga tulevad kaasa kolm tasuta disainimalli: Beez, JA Purity ja MilkyWay. Järgnev joonis annab ülevaate MilkyWay kataloogi sisust (vt Joonis 4).

Joomla / templates / rhuk_milkyway /		
Name	Type	Size
css	File folder	
html	File folder	
images	File folder	
component	PHP Script	2 KB
favicon	Icon	2 KB
index	Chrome HTML Document	1 KB
params	Configuration settings	1 KB
template_thumbnail	Adobe Fireworks PNG File	8 KB
templateDetails	XML File	5 KB
index	PHP Script	6 KB

Joonis 4. MilkyWay disainimalli kataloog

Ülevaade veel tutvustamata failidest MilkyWay disainimalli kataloogi näitel:

- **html kaust** - Sisaldab HTML ümberkirjutisi (*Overrides*).
- **component.php** - Defineerib loogika, mille põhjal kuvatakse printerisõbralikud lehed.
- **favicon.ico** - 16x16 pixlit suur ikoon, mis kuvatakse veebilehitseja aadressiribal.
- **index.html** – Rakendatakse juhul kui *index.php* pole kättesaadav. Sisaldab ainult valget taustavärvi.
- **params.ini** - Fail kuhu salvestatakse disainimalli parameetrid.

- **template\_thumbnail.png** – Disainimalli *thumbnail* ehk eelvaate pilt. Antud faili suurus 206x150 pixlit ja formaadiks PNG. Thumbnail kuvatakse administratiivliideses.

## 2.4 Anatoomia

Disainimalli tuuma moodustavad kaks faili: *index.php* ja *templateDetails.xml*. Järgnevalt ülevaade mõlemast koos näidetega.

### 2.4.1 index.php

Disainimalli südameks on PHP formaadis fail, mille koostise moodustab HTML vorming põimituna PHP ja Jdoc lausetest. HTML koodi abil määratakse lehekülje ülesehitus, PHP lisab loogika sisu dünaamiliseks kuvamiseks ning Jdoc laused märgistavad positsioonid, kus kuvatakse sisu. (Sevardia & Crowder, 2010)

Järgnevalt näide *index.php* koodist (vt Koodinäide 1).

```
<?php
// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );
?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo $this->language;
?>" lang="<?php echo $this->language; ?>" >
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="<?php echo $this->baseurl
?>/templates/system/css/system.css" type="text/css" />
<link rel="stylesheet" href="<?php echo $this->baseurl
?>/templates/system/css/general.css" type="text/css" />
<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
$this->template ?>/css/template.css" type="text/css" />
<!--[if IE]>
    <link rel="stylesheet" type="text/css" href="<?php echo $this->baseurl
?>/templates/<?php echo $this->template ?>/css/template_ie.css" />
<![endif]-->
</head>
<body>
<div id="mainHeader">
    <div id="header">
        <div id="sitename"><a href="<?php echo $this->baseurl ?>"
alt="<?php echo
```

```

$this->title; ?>" ><?php echo $this->title; ?></a></div>
    </div>
        <div id="rightHeader"><jdoc:include type="modules" name="topmenu"
style="xhtml"/></div>
</div>
<div id="mainContent">
    <jdoc:include type="message" />
    <jdoc:include type="component" />
</div>
<div id="mainFooter">&copy;<?php echo date("Y"); ?> <?php echo $this->title; ?>Kõik
Õigused Kaitstud.</div>
</div>
</body>
</html>

```

*Koodinäide 1. index.php*

Ülevaade tähtsamatest komponentidest:

```

<?php
// no direct access
defined( '_JEXEC' ) or die( 'Restricted access' );
?>

```

Kontrollitakse, kas faili lugeda sooviv rakendus kuulub Joomla süsteemi alla. Kui ei kuulu siis blokeeritakse edasipääs.

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

```

Dokumendi tüübi deklareerimine, antud juhul XHTML 1.0 Transitional.

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo
$this->language; ?>" lang="<?php echo $this->language; ?>" >

```

Dokumendi keele valimine vastavalt Joomla sätetele.

```

<jdoc:include type="head" />

```

Jdoc lause, mis lisab päise. Päis koosneb lehekülje meta-andmetest (vt Koodinäide 2)

```

<meta http-equiv="content-type" content="text/html; charset=utf-8" />
<meta name="robots" content="index, follow" />
<meta name="keywords" content="joomla, Joomla" />
<meta name="description" content="Joomla, the ultimate CMS." />
<meta name="generator" content="Joomla! 1.5 - Open Source Content Management" />
<title>Veebilehe tiitel! </title>
<link href="/templates/disainimall/favicon.ico" rel="shortcut icon" type="image/x-
icon" />
<link href="/veebileht.ee/index.php?format=feed&type=rss" rel="alternate"
type="application/rss+xml" title="RSS 2.0" />
<link href="/veebileht.ee/index.php?format=feed&type=atom" rel="alternate"
type="application/atom+xml" title="Atom 1.0" />
<script type="text/javascript" src="/media/system/js/mootools.js"></script>
<script type="text/javascript" src="/media/system/js/caption.js"></script>

```

*Koodinäide 2. Päis*

```

<link rel="stylesheet" href="<?php echo $this->baseurl
?>/templates/system/css/system.css" type="text/css" />

```

```

<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/system/css/general.css"
type="text/css" />

```

```

<link rel="stylesheet" href="<?php echo $this->baseurl ?>/templates/<?php echo
$this->template ?>/css/template.css" type="text/css" />

```

Dünaamilised lingid stiililehetele, esimesed kaks viidet on *system* disainimalli CSS failidele, mis rakendatakse süsteemi- ja veateadete kuvamisel. *System* disainimall asub kataloogis *templates/system/* ning üldtuntud tavana lisatakse antud kaks stiililehte igale disainimallile. Kolmas viide on antud disainimalli enda CSS stiililehele.

Ülevaade PHP meetoodikast:

```

<?php echo $this->baseurl ?>

```

Esimene meetod adresseerib Joomla lähtekataloogi.

```

<?php echo $this->template ?>

```

Teine meetod adresseerib antud disainimalli. Peale lehekülje genereerimist näeb kolmas viide välja selline:

```
<link rel="stylesheet"
href="http://www.veebileht.ee/templates/disainimalli_naide/css/templ
ate.css"
type="text/css" />
```

Dünaamiline linkide kuvamine lubab disainimalli kasutada korduvalt, mitme lehe genereerimisel, ilma viiteid käsitsi muutmata.

```
<!--[if IE]>
<link rel="stylesheet" type="text/css" href="<?php echo $this-
>baseurl
?>/templates/<?php echo $this->template ?>/css/template_ie.css" />
<![endif]-->
```

Kui lehekülje kuvamiseks kasutatakse Internet Explorer veebilehitsejat siis adresseeritakse ka *template\_ie.css* stiilileht.

```
<div id="sitename">
    <a href="<?php echo $this->baseurl ?>" alt="<?php echo
$this->title; ?>" ><?php echo $this->title; ?></a></div>
</div>
```

Kaks PHP meetodit, esimene adresseerib Joomla alamkataloogi ja teine kuvab lehekülje pealkirja. Peale lehekülje genereerimist näeb kood välja selline:

```
<div id="sitename">
<a href="http://www.veebileht.ee" alt="Disainimalli Näide"
>Disainimalli Näide</a>
</div>
```

```
<div id="rightHeader">
<jdoc:include type="modules" name="topmenu" style="xhtml"/>
</div>
```

Jdoc lause, mis lisab mooduli positsiooni. Positsiooni nimi on *topmenu* ja stiiliks *xhtml*.

```
<jdoc:include type="message" />
```

Süsteemi- ja veateadete kuvamine.

```
<jdoc:include type="component" />
```

Veebilehe sisu kuvamine.

```
<div id="mainFooter">&copy;<?php echo date("Y"); ?> <?php echo
$this->title; ?>Kõik Õigused Kaitstud.</div>
</div>
```

PHP meetodid aasta ja lehekülje pealkirja lisamiseks. Peale lehekülje genereerimist näeb kood välja selline:

```
<div id="mainFooter">
&copy;2010 Disainimalli Näide Kõik Õigused Kaitstud.</div>
</div>
```

Antud disainimall koosneb kolmest peamisest DIV elemendist: *mainHeader*, *mainContent* ja *mainFooter*. Lisaks on *mainHeader* piiride sees veel *header*, *sitename* ja *rightHeader*. Kokku on neli Jdoc lauset, mille tüübid on: *head*, *modules*, *message* ja *component* ja mis on paigutataud vastavalt illustreerivale näitele (vt Joonis 5).

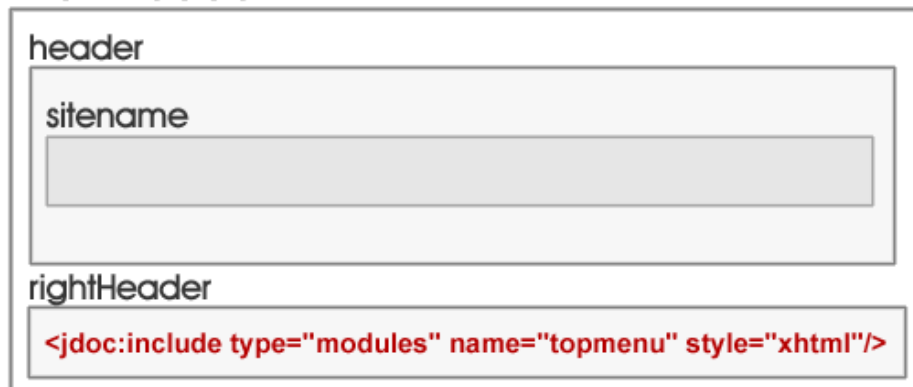
<head>

<jdoc:include type="head" />

</head>

<body>

mainHeader



mainContent



mainFooter



</body>

Joonis 5. *index.php* ülesehitus

## 2.4.2 templateDetails.xml

Fail *templateDetails.xml* koosneb andmetest, mis on vajalikud disainimalli paigaldamiseks ja käsitlemiseks. Paigaldamisel kopeerib Joomla nimekirja põhjal failid ja asetab need vastavalt etteantud skeemile. Kui disainimall on lisatud, loeb Joomla andmete põhjal moodulite asukohad ja parameetrid ning kuvab need administratiivliideses.

Järgnevalt näide *templateDetails.xml* faili sisust (vt Koodinäide 3).

```
<?xml version="1.0" encoding="utf-8"?>
```

```

<!DOCTYPE install PUBLIC "-//Joomla! 1.5//DTD template 1.0//EN"
"http://dev.joomla.org/xml/1.5/template-install.dtd">
<install version="1.5" type="template">
  <name>Disainimalli nimi</name>
  <creationDate>27/12/10</creationDate>
  <author>Autori nimi</author>
  <authorEmail>autor@email.ee</authorEmail>
  <authorUrl>http://www.autor.ee/</authorUrl>
  <copyright>©2010 Autor</copyright>
  <license>GNU/GPL </license>
  <version>1.0.0</version>
  <description>Disainimalli kirjeldus</description>
  <files>
    <filename>templateDetails.xml</filename>
    <filename>index.php</filename>
    <filename>template_thumbnail.png</filename>
    <filename>css/template.css</filename>
    <filename>images/logo.gif</filename>
    <filename>images/house.gif</filename>
    <filename>images/footer.gif</filename>
  </files>
  <positions>
    <position>top</position>
    <position>left</position>
    <position>footer</position>
  </positions>
</install>

```

*Koodinäide 3. templateDetails.xml*

Antud näites on disainimall, mis koosneb seitsmest failist, millest kolm on pildid. Mooduli positsioone on määratud kolm: *top*, *left* ja *footer*.

## 2.5 Jdoc laused

Jdoc lauseid kasutatakse disainimalli ülesehitusse sisu lisamiseks. Mis liiki sisu edastatakse, sõltub parameetrist *type*, mis on iga lause sees defineeritud. Positsioon sõltub, kus HTML vormingus lause täpsemalt asub. (Joomla! Jdoc, 2010)

Jdoc süntaks on defineeritud Joomla süsteemi poolt ja on mõeldud asendada PHP programmeerimiskeelt, eesmärgiga lihtsustada disainimallide loomist. Arendajate seisukohast vaadates võib see aga vastupidi toimida, muutes koodipildi ja arusaama süsteemi toimimisest kohati keerulisemaks.



Joomla süsteemi poolt on määratud järgmised Jdoc lausete tüübid:

### Komponent

```
<jdoc:include type="component" />
```

Kui konkreetset komponenti pole defineeritud siis loetakse automaatselt *com\_content* komponent, mis on veebilehe peamise sisu edastamiseks ehk tavaliselt koht, kus asub veebilehe üldine tekst (Joomlas artiklid). Antud tüüp võib esineda ainult üks kord terve lehe peale.

### Päis

```
<jdoc:include type="head" />
```

Päise lisamine. Päis koosneb meta-andmetest, lisanduvatest skriptidest ja stiililehtedest.

### Sõnum

```
<jdoc:include type="message" />
```

Väljund süsteemi- ja veateadetele. CSS stiilimustrid, mille põhjal süsteemi- ja veateated kuvatakse, asuvad *templates\system\css\system.css*

### Installatsioon

```
<jdoc:include type="installation" />
```

Kasutatakse Joomla paigaldamisel serverisse, kuvab installatsiooniprotsessi.

### Moodul

```
<jdoc:include type="module" name="mainmenu" title="Main Menu" />
```

Kasutatakse automaatselt ühe ettenäidatud mooduli kuvamiseks, mis määratakse atribuutidega *name* ja *title*.

Antud näites on disainimalli positsioon, mis on mõeldud ainult *mainmenu* moodulile, mille nimi on *Main Menu*. Lehe genereerimisel kuvatakse automaatselt *mod\_mainmenu* moodul. Kui antud nime ja pealkirjaga moodulit ei leita, siis ei kuvata midagi.

## Moodulid

```
<jdoc:include type="modules" name="title" />
```

Kasutatakse mitme mooduli lisamiseks (võib lisada ka ainult ühe), seejuures pole mooduleid kindlalt määratud. Tegemist on moodulite kuvamiseks mõeldud positsiooniga, kuhu saab soovitud mooduleid määrata/muuta administratiivliideses.

Kui antud mooduli positsioonile on määratud mitu moodulit siis kuvatakse need üksteise järel. Järjekorda saab muuta *Template Manager* liideses.

Atribuudiga *name* määratakse mooduli positsiooni nimi. Nimi on soovitatav panna selline, mis iseloomustaks kohta, kus lehel moodulit kuvama hakatakse, nt *left* või *footer*.

Moodulite positsiooni nimi lisatakse ka *templateDetails.xml* andmetesse *position* tagide vahele, ehk antud näite korral:

```
<position>title</position>
```

Segadust võib tekitada atribuut *name*, mis Jdoc lauses *module* tähendab kuvatava mooduli tüüpi, kuid lauses *modules* hoopis mooduli positsiooni nime.

Kõiki Jdoc lauseid võib nimetada kui „disainimalli positsioonideks”, mis määravad edastatava info tüübi ja positsiooni lehel. Viimasena lahatud Jdoc lauset võib täpsemalt nimetada „mooduli positsiooniks”, mis on mõeldud ainult moodulite kuvamiseks.

## 2.6 Parameetrid

Parameetreid kasutatakse disainimalli välimuse muutmiseks administratiivliideses, seejuures koodipilti käsitsi muutmata. See on hea, kui kasutaja soovib muuta disainimalli välimust, kuid samas pole tal piisavalt teadmisi, et muuta *index.php* koodi. Eriti levinud on see võimalus tasuliste disainimallide seas. (Shreves, 2010)

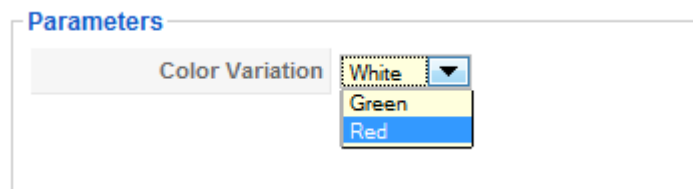
Järgnevalt näide lihtsamast disainimalli parameetrist:

```
<params>
  <param name="colorVariation" type="list" default="white"
  label="Color Variation" description="Color variation to use">
    <option value="red">Red</option>
```

```
<option value="green">Green</option>
<option value="white">White</option>
</param>
</params>
```

Kõik parameetrid lisatakse *myTemplate.xml* nimestikku *params* tagide vahele. Antud näites on parameeter, mille nimi on *colorVariation*. Tüübiks on *list* ehk administratiivliideses kuvatakse valikuvõimalused listina (vt Joonis 6). Algväärtus on *white*, sildiks on *Color Variation* ja parameetrit iseloomustavaks tekstiks „*Color variation to use*”.

Ülevaate erinevatest parameetri tüüpidest leiab Joomla dokumentatsioonist. ([http://docs.joomla.org/Tutorial:Template\\_parameters](http://docs.joomla.org/Tutorial:Template_parameters))



Joonis 6. Parameetri kuvamine listina (Template Manager)

Parameetri väärtus salvestatakse faili *params.ini*, mis asub disainimalli alamkataloogis. Kui fail on tühi siis salvestab Joomla automaatselt sinna algväärtuse. Antud näite korral on *params.ini* sisu järgmine:

```
colorVariation=white
```

Ning parameetri rakendamine *index.php* koodis:

```
<link rel="stylesheet" href="<?php echo $this->baseurl
?>/templates/rhuk_milkyway/css/<?php echo $this->params-
>get('colorVariation'); ?>.css" type="text/css" />
```

Juhul kui kasutaja valib värvi *white*, lisatakse lehele *white.css* stiilileht.

## 2.7 Mooduli positsiooni stiilid (*Modul Chrome*)

Igale mooduli positsioonile on võimalik lisada atribuut *style*, mis määrab kuidas Joomla moodulit kuvab. Näiteks alljärgneval mooduli positsioonil, nimega *topmenu*, on määratud *style* atribuudi väärtuseks *xhtml*, mis lisab mooduli ümber DIV tag-i (või teistpidi öeldes, moodul kuvatakse DIV tag-i sees). (Joomla! Chrome, 2010)

```
<jdoc:include type="modules" name="topmenu" style="xhtml"/>
```

Joomla süsteemi poolt on defineeritud järgmised mooduli positsiooni stiilid:

- ***none*** – Mooduli ümber ei lisata midagi. Kasutatakse vaikimisi kui atribuuti pole lisatud.
- ***table*** – Mooduli ümber kuvatakse tabel (*HTML table*).
- ***rounded*** – Mooduli ümber lisatakse ümarate nurkadega DIV tag.
- ***horz*** – Mooduli ümber lisatakse kaks tabelit.
- ***xhtml*** – Mooduli ümber kuvatakse DIV tag.
- ***outline*** – Mooduli ümber lisatakse punktiirjoontega DIV tag.

## 2.8 HTML ümberkirjutised (*HTML overrides*)

HTML ümberkirjutistega saab muuta disainimallis kuvatava sisu väljanägemist. Kui mooduli positsiooni stiilidega (*Modul Chrome*) sai muuta ümbrist, kus moodul kuvatakse, siis HTML ümberkirjutised võimaldavad muuta mooduli väljanägemist täielikult.

Loogika, mille põhjal ümberkirjutised töötavad, vastab MVC mudelile (*Model View Controller*), mis tähendab, et kõikide Joomla laienduste sisu ja välimus hoitakse üksteisest lahus. (Sevardia & Crowder, 2010)

Näiteks kui disainimallis on määratud mooduli *mod\_login* välimus ning antud moodulit ka disainimallis kuvatakse, siis rakendatakse disainimalli poolt kehtestatud välimus, mitte see, mis *mod\_login* mooduliga endaga kaasa tuleb.

Kõik ümberkirjutised asuvad kaustas */html*.

## 3. OÜ Antiigihunt disainimalli loomine

Antud rakenduse arendus koosnes viiest peamisest sammust: planeerimine, graafilise kavandi loomine, kodeerimine, sidumine ja testimine. Peale viimasele sammule jõudmist järgnes jälle esimene, sest plaani tuli muuta ning nii hakkas ring jälle otsast peale. Toimus lugematu arv kordusi, mille järel valmis viimane tõsiseltvõetavaim versioon. Kuna graafiline disainer, kodeerija, Joomla spetsialist ja klient olid ühes isikus, siis ei tekkinud probleeme ei tööjaotuses ega kliendi mõistmises.

### 3.1 Kriteeriumid

Osaühing Antiigihai nõuded veebirakenduse loomiseks:

- Veebileht koosneb neljast leheküljest:
  - Ettevõtet tutvustav lehekülg, mis kuvatakse avalehena.
  - Tooteid tutvustav lehekülg.
  - Teenuseid tutvustav lehekülg
  - Kontaktlehekülg.
- Veebileht on kolmekeelne, keeled on eesti, inglise ja soome. Igal hetkel on võimalus keeli vahetada.
- Lisanduvad rakendused nagu otsingumootor, uudistetahvel, küsitlus- ja kontaktvorm.
- Veebilehel kuvatakse OÜ Antiigihunt logo.
- Igal leheküljel on põhilised kontaktandmed kättesaadavad.

Ülevaade ettevõtte tegevusest ja imagost:

Veebilehe tellijaks on Osaühing Antiigihunt, mis on mainekas äriettevõtte, mille põhitegevuseks on antiikmööbli müük ja restaureerimine. Eestis on tegutsenud juba üle viie aasta, lähituleviku plaaniks on kanda kinnitada ka Soomes. Ettevõtte omab kasvavat klientuuri ning on tuntud oma hea kvaliteedi poolest. Firma edendab rohelist mõtteviisi ja kasutab ainult kõige loodusõbralikemaid tehnoloogiaid. Kliendi sooviks jäi disaini realiseerimisel kasutada läbivat pruuni värvi, mis iseloomustaks nende küllaldast kokkupuudet puiduga.



Joonis 7. OÜ Antiigihunt logo

## 3.2 Planeerimine

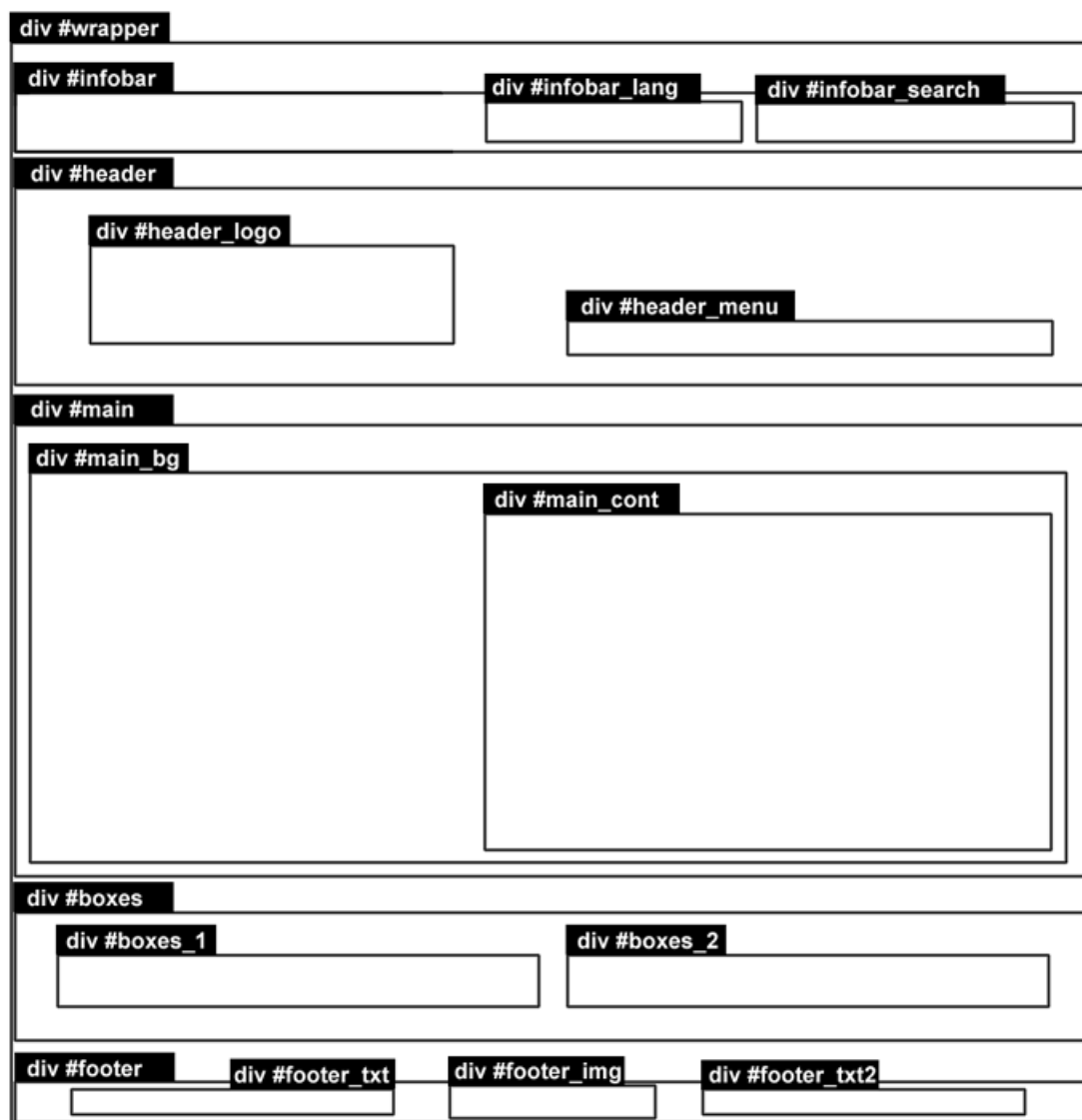
Valmis esmane eskiis lehekülje ülesehitusest – logo, teksti ja navigatsiooni paiknemine. Tekkis ettekujutus, kus peaks paiknema sisukomponent, kaks moodulit rakenduste jaoks, menüü- ja otsingumoodul. Kasutatud vahenditeks paber ja pliiats.

## 3.3 Graafilise kavandi loomine

Programmiga Adobe Photoshop valmis graafiline kavand leheküljest, millest sai välja lõigatud järgnevad pildid: *logo.png*, *searchbg.gif*, *menubg.png*, *bg.gif*, *footerimg.png*.

## 3.4 Kodeerimine HTML-i

Graafilise kavandi põhjal toimus HTML vormi loomine, keskkonnaks Adobe Dreamweaver. Kokku tuli 17 DIV tagi, millest igäüks sai CSS stiilimustri vastavate sätetega. Lehe moodustavad viis peamist sektsiooni: *#infobar*, *#header*, *#main*, *#boxes* ja *#footer* (vt Joonis 8).



Joonis 8. DIV ülesehitus

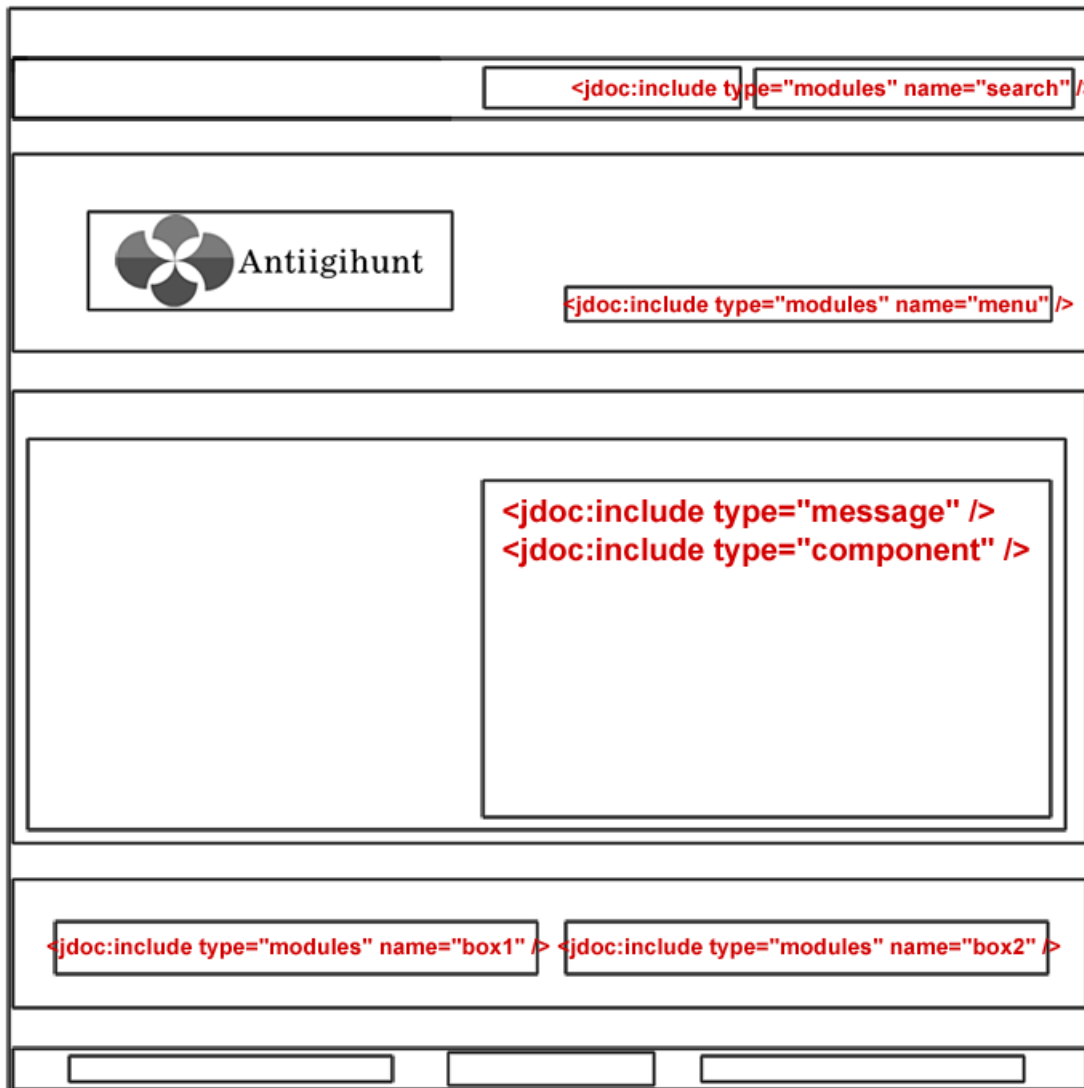
### 3.5 Sidumine Joomla!ga

Koodi lisamine toimus Adobe Dreamweaver-iga, kus *index.html* fail sai hiljem ümber salvestatud PHP formaati. Kokku tuli seitse Jdoc lauset, millest kolm lisavad automaatselt päise, süsteemiteadete väljundi ja sisukomponendi. Ülejäänud neli on mooduli positsioonid, kuhu määratakse moodul vastavalt soovile.

Antud juhul tuleb mooduli positsioonile nimega *search* otsingumoodul, nimega *menu* navigatsioonimoodul ja kahte viimasesse, nimedega *box1* ja *box2*, vastavalt lehele kas uuistetahvel, küsitlustik või kontaktvorm (vt Joonis 9).

Ühelegi mooduli positsioonile pole määratud *style* atribuuti, kuna kõik tulevased moodulid kuvatakse juba DIV tagide sees, mis on kergesti muudetavad.

`<jdoc:include type="head" />`



Joonis 9. Sisü paiknemine



## 3.6 Testimine

Lehekülg sai testitud kolme kaasaegse (Google Chrome 8, Mozilla Firefox 3.6 ja Internet Explorer 8) ja ühe vanema põlvkonna veebilehitsejaga (Internet Explorer 6).

Uuemad veebilehitsejad said kõik ideaalselt hakkama ja tulemus oli täiesti identne. IE6 puhul tekib probleeme PNG pildifailide kuvamisel, nimelt IE6 ei toeta piltide läbipaistvust (*transparency*). Lisaks mõningased anomaaliad CSS kuvamisel, kuid samas leht funktsioneerib täielikult ja info on kättesaadav.

Disainimalli peamine probleem on sisu kuvamine. Nimelt ruum, mille piires saab sisukomponenti kuvada, on fikseeritud ja juhul kui seda ületatakse, liigub kujundus paigast. Lahendus probleemile oleks standartsem lehe ülesehitus, näiteks sisu keskel ja lisanduvad moodulid äärtes.

Minimaalne resolutsioon lehe vaatamiseks on 1024x 768 pikslit. Juhul kui resolutsioon on väiksem siis ei kuvata lehte horisontaalselt enam täielikult (veebilehitsejasse tekivad ka horisontaalsed kerimisribad).

Võib väita, et antud paarileheküljelise ja lihtsa struktuuriga disainimalli valmistamine Joomla süsteemiga on igati edukas, kuid kogemuste puudumisel nõuab see väga suurt pühendumist. Kurssi on vaja viia ennast rakenduse terminoloogia ja toimeloogikaga, mis võtab omajagu aega.

## Kokkuvõte

Käesolevas seminaritöös koostas autor ülevaate Joomla disainimalli tööpõhimõtetest ja võimalustest koos näidetega.

Uurimustöö tulemusena valmis äriettevõtte disainimall Joomla sisuhaldussüsteemile, mis on kättesaadav aadressil <http://pollex.planet.ee/antiigihunt/>. Antud rakendus funktsioneerib täielikult etteantud stsenaariumite piires.

Autor veendus Joomla süsteemi paindlikkusest, mis lubab valmistada disainimallile vastavalt ka kõige „loomingulistema” soovide korral. Uurimustöö põhjal saab kinnitust väide, et kõik HTML kujundusmallid, mis on kuvatavad veebilehitsejates, on seotavad ka Joomla sisuhaldussüsteemiga.

Negatiivse poole pealt võib välja tuua Joomla vähese arendajasõbralikkuse. Võimalused disainimallide loomiseks on küll paindlikud, kuid kui peaks tekkima vajadus arendustöödeks on olemasoleva dokumentatsiooni põhjal raske midagi muuta.

# Kasutatud kirjandus

- Harwani, B, M. (2009). *Foundation Joomla!*. New York: Apress.
- Joomla!. (2010). *What's a content management system?*. Loetud 20.08.2010 aadressil <http://www.joomla.org/about-joomla.html>
- CMSWire. (2009). *2009 Open Source CMS Market Share Report*. Loetud 12.08.2010 aadressil <http://www.cmswire.com/downloads/cms-market-share/>
- GNU. (2010). *GNU General Public Licence*. Loetud 20.08.2010 aadressil <http://www.gnu.org/licenses/gpl.html>
- Lanham, C., & Kennard, J. (2010). *Mastering Joomla! 1.5 Extension and Framework Development*. Birmingham, UK: Packt Publishing.
- Severdia, R., & Crowder, K. (2010). *Using Joomla*. Sebastopol, USA: O'Reilly.
- Rhamel, D. (2009). *Beginning Joomla! Second Edition*. NY, USA: Apress.
- Shreves, R. (2010). *Joomla! Bible*. Indianapolis, USA: Wiley Publishing.
- Joomla! Jdoc (2010). *Jdoc statements*. Loetud 19.12.2010 aadressil [http://docs.joomla.org/Jdoc\\_statements](http://docs.joomla.org/Jdoc_statements)
- Joomla! Chrome. (2010). *What is module chrome?*. Loetud 27.12.2010 aadressil [http://docs.joomla.org/What\\_is\\_module\\_chrome%3F](http://docs.joomla.org/What_is_module_chrome%3F)