



# **Technical Summary of JK Design's bespoke CMS**

(Aka: Administrative Interface)

**CMS Version 2.1 rev. 1**

[www.jkdesign.com](http://www.jkdesign.com)

May 23, 2011

## ARCHITECTURE

### Architecture Overview

When developing a web site, JK Design employs templates and a database-driven architecture using the Model-View-Controller (also known as **MVC**) development pattern, which consists of Model – Database query calls, View – HTML display, and Controller – Application logic. We use MySQL, MS SQL Server, or Oracle databases to store page content and related information. PHP is used for the site's logic.

JK Design uses a powerful, open-source PHP framework with a small footprint, called **CodeIgniter** - <http://codeigniter.com/>, for MVC-based website application development. This architecture allows us to separate the site's design from the logic, as well as standardize and streamline code development.

A more detailed technical documentation on the CodeIgniter framework can be found at their User Guide: [http://codeigniter.com/user\\_guide/](http://codeigniter.com/user_guide/).

### Database (MySQL, MS SQL Server, Oracle)

At the most basic level, a web site consists of a number of pages. Naturally, we store these pages in the database. These records, or “pages,” consist of a number of components and field, and here are the components of interest:

- **pages** – This table holds the actual page records.
  - **page\_id** – This is a unique identifier of the record in the table. This is used in the CMS tool when performing actions on a page. It's also used when creating relationships between pages and other related tables.
  - **parent\_id** – This is a numeric value that associates each page with another page that is up a level in hierarchy (i.e. the parent of this page). If the value is 0, the page is at the top level.
- **page\_content** – This table holds the content or data portion of the page records.
  - **slug** – This is a string value with only lower case letters, numbers and hyphens. This allows us to create a normalized URL that is also search engine friendly: <http://www.yoursite.com/about-us/our-staff/> “about-us” and “our-staff” are slugs. See the section on URL Rewriting under “Controller” below for an explanation of this technique.
  - **title** – This is the title of the page. It's generally used in an H1 tag on the page itself as well as in any menu and in the title bar.
  - **description** – This contains the copy text of the page with HTML formatting. We use a WYSIWYG editor in the CMS tool to help in the creation of this HTML.
- **navigation\_positions** - This table holds the navigational elements of the website, which typically include the Main Navigation, Header Navigation, and Footer Navigation
- **pages\_to\_navigation\_positions** – This table holds the multi-to-multi association between pages and navigation\_positions, in order to construct the website's navigation menus.
  - **display\_order** – A numeric value that indicates the display order of the page records in each navigation menu.

There may be other fields used in the site to determine page's active status and specific meta tags, etc. Additional tables are generally included in the database, such as for storing contact submissions, corporate locations, or representative information.

Database queries are handled by the Model component of the MVC structure.

### **Rendering** (XHTML / CSS)

HTML is the language of the Internet. It's important in both displaying information to the end user as well as making information available to search engines for easy indexing. We write all of our sites in XHTML which is a strict form of HTML that is XML-compliant. This makes it easier for browsers and search engines to parse each page decreasing load time and increasing search engine traffic.

In order to keep the XHTML as simple and clean as possible, we use Cascading Style Sheets (CSS) to define and position design elements. Without CSS, an XHTML page would simply be a flow of tags and content in a generic style that is determined by the browser. With CSS, the pages are formatted logically for the end user and styled appropriately.

The "Views" component of the MVC structure drives the display of HTML, separating core PHP logic from HTML code. This allows rapid initial development as well as the ability to change design elements without touching the core logic.

### **Logic** (PHP)

PHP is already one of the top scripting languages used on web sites. JK Design uses this free, yet powerful language to quickly develop the logic necessary to display the information entered into the database. PHP is used to connect the database with the template and handle the page logic, which ranges from figuring out what records to pull from the database to validating the input on a contact form.

PHP logic is handled by the Controller component of the MVC structure.

### **URL Rewriting**

Static URLs are handled by default by the CodeIgniter framework, which are designed to be search-engine and human friendly. Rather than using the standard "query string" approach to URLs that is synonymous with dynamic systems, CodeIgniter uses a segment-based approach:

<http://www.yoursite.com/segment1/segment2/segment3/segment4/>

Each section is optional, however, section 4 cannot exist without 3, and 3 cannot exist without 2. As such, 2 cannot exist without 1. Each section is made up of any combination of letters, numbers, or a dash. These section strings are known as slugs. The index.php file then handles the page accordingly.

## DEVELOPER DOCUMENTATION

For clients who require a database driven PHP website, JK Design implements an easy to use and real-time web front end to manipulate that database, backed by the CodeIgniter Framework along with its set of powerful functionality libraries. This front end consists of a basic set of PHP classes and functions that we generally tailor to fit the needs of each individual client. Each front end created for a given client is capable of supporting multiple levels of access all of which are password protected from the general public. These access levels are defined at the beginning of the project. Once completed, the client will have the ability to change virtually any element of his website that is created with information from his database.

### Core Classes

**Filename:** Core.php

**Author:** JK Design

**Type:** Core Library

**Desc:** This file is a class declaration used for initiating the application, and calls upon other classes/functions that are necessary for the website to function.

**Filename:** Pages.php

**Author:** JK Design

**Type:** Core Library

**Desc:** This file is a class declaration used for displaying the Pages of the website.

**Filename:** Users.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for handling the User-related functionality of the website, such as site administration login, user registration, and forgotten password reset.

**Filename:** Form\_generator.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for generating HTML forms, along with client-side and server-side validation.

**Filename:** Form\_submission.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for handling form submissions, and subsequent actions such as email notifications.

**Filename:** Languages.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for handling the website's languages, allowing for optional multiple language support on standard pages.

**Filename:** MY\_Form\_validation.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for extending the CodeIgniter's native Form Validation class, in order to facilitate additional form validation functionality used by the website.

**Filename:** Application\_message.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for displaying messages relating to the application's functionality, such as confirmation messages on successful page edits.

**Filename:** Debug.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for displaying debug information used by JK Design developers logged into the website, in order to troubleshoot any issues within the website, without disturbing the website's normal use.

**Filename:** Application\_message.php

**Author:** JK Design

**Type:** Core Library

**Desc:** Class for displaying messages relating to the application's functionality, such as confirmation messages on successful page edits.

## **CKeditor**

CKeditor (<http://www.ckeditor.net>) provides a GUI interface for editing HTML via a Web based interface. It has many features that can be used, which are similar to features in Word. To see an example of CKeditor in action, with all available features, see: <http://ckeditor.com/demo>

You will see about 80 buttons that can be used for various features. However, implementing all of these features in our CMS tool, is overkill. In fact, the more features that are implemented, the more our Clients mess up the look of their Web sites (i.e.: they may add font faces and colors that do not go with the rest of their site's design). Because of these issues, JK Design will typically only implement the following features, within the CMS interface:

Source, Paste as Plain Text, Paste from Word, Bold, Italics, Underline, Strike Through, Subscript, Superscript, Insert/Remove Numbered List, Insert/Remove Bulleted List, Decrease Indent, Increase Indent, Insert/Edit Link, Remove Link, Insert/Edit Anchor, Insert/Edit Image, Insert Special Character, Maximize the Editor Size.

Other options (e.g., Font, Size, Text Color, etc.) will NOT be made available in order to maintain a consistent look and feel throughout the site.

### **Common configuration:**

- Place a ckeditor folder within includes folder (/js/third-party/ckeditor/)
- Use the following code to initialize the editor (within the Display\_Form() function) in the admin section:

```
$this->admin->display_form_ckeditor(  
array('field'=>'description','name'=>'Description:',  
'value'=>$this->record_values['description'],'rows'=>5,'cols'=>25,  
'readonly'=>$this->record_readonly)),
```

## **Installation**

### **Step 1: Design Database**

This is the first step after the website is graphically designed and the html created. From here a developer will work with the client and project manager to discuss the types of information that

will need to be stored in the database, and what elements of the website will need to be editable. The developer will then design and create the database to those needs.

Generally our databases consist of several tables, all identified by a descriptive name (i.e.: product\_categories). The columns in these tables are then identified by a descriptive name with the table name as its leader (i.e.: product\_category\_id). The primary key of the table is generally always an auto incrementing integer.

### Step 2: Develop Front End

This step assumes the successful complete of step 1 and involves the moving of our class/function files to the development server. From here the developer in collaboration with the project manager will write a front end interface that allows for the manipulation of any of the user editable elements of the database. The CMS generally follows this flow: *select section -> select/add row -> edit/populate row -> save*. Once the data is saved, the CMS writes to the database and the website is therefore updated in real time.

### Step 3: Develop the website itself

This function assumes the successful completion of step 1 and at least partial completion of step 2. From here the developer takes a copy of the JK Design CMS code, add his/her raw HTML page, and starts breaking it up into PHP that allows for the display of any of the dynamic information pulled from the database.

### Example

In this example we will assume that we have a client that simply wants to have a database front end to news driven website based on news categories. We could create the database like this:

#### “News”

news_id	int(11)	PRI unsigned auto_increment
title	varchar(255)	
description	text	
image	varchar(255)	
author_id	int(11)	
category_id	int(11)	

Note: In the above example author\_id represents a foreign key from the “Authors” table below. By naming the local fields the same as their foreign counterparts, you can use the MySQL “USING” keyword to simplify SQL statements –  
 SELECT \* FROM news LEFT JOIN authors ON news.author\_id = authors.author\_id

#### “Authors”

author_id	int(11)	PRI unsigned auto_increment
first_name	varchar(255)	
last_name	varchar(255)	
email	varchar(255)	

#### “Categories”

category_id	int(11)	PRI unsigned auto_increment
title	varchar(255)	

Once the database is created, the developer creates his CMS. The user will have the ability to create or edit a news item. If he is creating a news item, the “author” field would be automatically filled in based on the user’s login and the categories field would be a drop down list that is automatically generated from information in the Categories table.

Once the CMS is completed, the developer would then write his PHP code into the site design to display the news items on the client's front page. He then passes the CMS section on to the client to become familiar with.

## **Deployment**

JK Design Web site deployment generally consists of four main stages:

1. Deployment of the code
2. Deployment of the database
3. Protection of the CMS and editing site configuration and security settings
4. Editing the configuration to work on the live server, such as changing the configuration paths and providing the application with the database credentials or DSN.

Security Features:

- JK Design utilizes a central settings file, located outside of the Web root for security purposes.
- The CMS is password protected.
- Form input is sanitized and validated to prevent malicious input

## **cmsdemo.jkdesign.com**

JK Design created a demo site at <http://cmsdemo.jkdesign.com>, which is used to demonstrate to potential clients, how the CMS can be used to edit client content. The actual CMS can be reached at <http://cmsdemo.jkdesign.com/admin> (user: guest@jkdesign.com pwd: guest1)

All client CMS look similar to the demo site, although the actual menu options for editing various client content will differ, depending on the type of content and the format of the actual page on the client's site.

## **Advanced CMS**

JK design can offer CMS tools to enable the editing and previewing of site prior to publishing to the public site. An Advanced CMS can provide custom work-flow requirements, such as different user account roles with privilege to approve records to be published, and or rolling back previous versions of the record.

The advanced CMS requires the use of two separate websites, though they can be located on the same server. One website is used for the public site and the other is used for entering content and performing previews.

## **Conclusion**

Our CMS is very scalable, and can adapt to even the most complex database manipulation means. From the smallest website such as the above example, or a large e-commerce site that requires many tables and complex SQL operations, our CMS will function as required by the client. The time involved in creating these interfaces is of course related to the complexity of the client's needs, however the majority of the time is spent writing the PHP into the site to display the finished product, rather than developing the interface itself. In other words, once the requirements are set, and the database created, the interface comes together very quickly and very efficiently.