

# Design Management System for talent.gamehouse.com Modification and Maintainability

Aditya Astono

Department of Informatics, Petra Christian University, Surabaya, Indonesia  
Information and Communication Technology, Fontys University of Applied Sciences, Eindhoven, The Netherlands  
aditya.astono@gmail.com

---

**Abstract.** Design is not a lasting component within a web application. There will come a time when the design is outdated and requires a new form of styling. However, in order to implement the new design, one or more developers are needed to modify the code of the application. The process will use a considerable amount of resources and time that might be more valuable to use for other more important cases. That is why the idea of creating a Design Management System (DMS) surfaced. DMS is created for talent.gamehouse website so that an individual with no knowledge in web development can alter the styling according to his/her preference. The application will also help developers who work on the website to decrease the time needed to do the design implementation. The process is done through several phases. The first phase is to inject a server-side programming language, such as PHP, Python, and Ruby to the Cascading Style Sheet, which is a presentation document that consists of styling properties for a website. The second phase is to create a database for the styling properties, so that the data is stored in a server instead of the CSS. The last and most crucial phase of the process is to create a function where a styling property can be modified using an SQL query. The query will alter the property to the user's liking. These phases are all generated through a front-facing site which the users will interact with. The site will be created using Hyper Text Markup Language (HTML), complemented with additional frameworks to enhance the user's experience. The application was tested through a usability session with one of the stakeholders of the application and generated a positive outcome for the fully-functional end result.

**Keywords:** CMS, Web Application, PHP, CSS, Design.

---

## 1. Introduction

GameHouse Europe is a leading publisher, developer and distributor of casual games worldwide. The company is a division of RealNetworks based on Seattle, USA. Popular casual game franchises, such as Delicious Emily's and Fabulous Angela were created by GameHouse Europe [1]. In order to retain the business growth and success, talented and passionate individuals working for the company will always be needed. That was when the company decided that a website which portrays the working culture, atmosphere, and upcoming events within the company was needed. The website will also be intended to attract talent across the globe to join GameHouse Europe. At the point, the website has already existed with several features that are useful to gain interest from prospectus. However, the company thought that the website was not yet completed and new features were needed for the website. The features needed for the website were not only customer-facing features, but also features that would help internal people to maintain the website. Maintainability is needed for content and design for the website. However, conventional websites require a sizable amount of both resources and time. The company would have to allocate several developers with the specific knowledge in web development to

perform maintenance for the website. The process would also take some time, which might hinder other important matters in the company that were possibly more important than doing the process. That is why a management system was needed for the website. A Content Management System (CMS) was created by the previous batch of developers and served to be an effective application to edit text inside the website. The application saved a lot of time required and it was used by several people in the company which had a little or no knowledge in web development. Yet, a management system for design properties did not exist, which meant a developer was needed to alter any sort of design within the website. Therefore, a design management system that would be able to provide the same functionality as the CMS was needed. The Design Management System (DMS) would slash such unnecessary resources and time to a minimum.

## 2. The Need for Design Management System

If a design alteration is required at some point in the future, there is only one approach to accomplish the goal. The approach is to change several values within the code that runs the website. The process will not require a significant amount of time. However, the case will only apply if the developer has a certain degree of

familiarity with the website. The other aspect that is of concern for the company is that some individuals with technical expertise will be the only ones who can do the implementation. So, not only it requires quite a bit of time, the process also needs some resources within the company which may have other more important obligations to do.

The situation is the reason why the idea surfaced at the first place. The idea was to create a simple and easy-to-use application that would alter design properties directly. The application was created with bearing non-technical people in mind. The application was created as simple as possible so that every employee inside the company could actually use the application without any prior specific knowledge.

### 3. System Design

#### 3.1. Use Case Diagram

The section will provide a use case diagram, as shown in Figure 1, which describes the procedure of using the DMS. After describing the general use case diagram, each main activity in the application will also be explained as activity diagrams. The diagram will split the flow to 2 different sides, which are View and System. View describes user-facing activities and System shows what the application does when an interaction is triggered. The diagrams are created by following the guidelines provided by Microsoft [2].

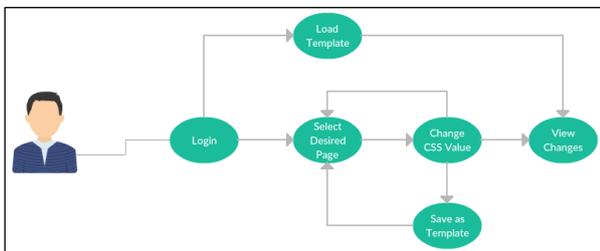


Figure 1. Use case diagram for DMS

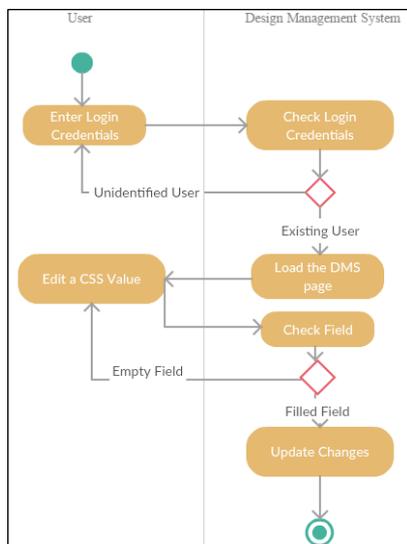


Figure 2. Activity diagram to change a CSS value

#### 3.2. Activity Diagram of Changing a CSS Value

The diagram shown in Figure 2 illustrates activities that should be done in order to change a design value of the website.

#### 3.3. Activity Diagram of Saving Changes to a Template

Figure 3 illustrates activities required to save the entire changes to a template.

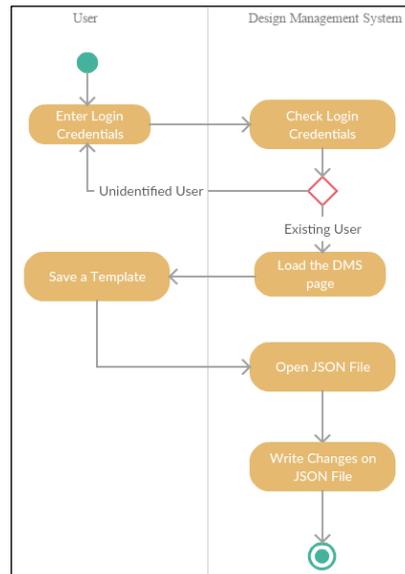


Figure 3. Activity diagram to save a template

#### 3.4. Activity Diagram of Loading a Template to the Website

The last activity diagram, shown in Figure 4, illustrates the working process when a user loads an existing template in the Design Management System.

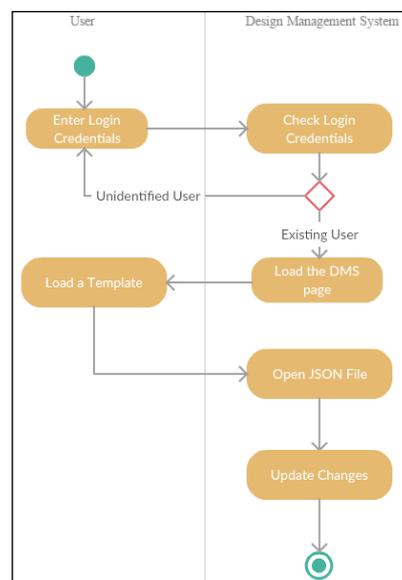


Figure 4. Activity diagram to load a template

## 4. System Implementation

### 4.1. PHP Integration to CSS

In order to dynamically change the values inside a CSS, a server-side programming language needs to be injected into the stylesheet document. In the project, PHP is chosen as the language to provide a connection between the CSS and database.

```
<?php
    header("Content-type: text/css; " .
          "charset: UTF-8");
?>
```

The injection is done by adding a CSS header on top of a .php file. By adding the header, web browser will run the PHP file as a stylesheet document. The advantage of injecting PHP in the document is the availability in using PHP variables and functions which helps web page to change content dynamically with the help of database implementation [3].

```
.-index-btn-custom {
    border-radius: 3px;
    background-color: <?=$cssValues[34]; ?>;
    color: white;
    text-align: center;
    padding: auto;
}
```

The section above illustrates how a data from the database is populated to the CSS file. The data stored in the database are pushed to an empty array. Then, the data need to be generated using PHP code which prints the values inside a CSS property.

### 4.2. Update a CSS Value

The update process is done through a form submission which uses Asynchronous JavaScript and XML (AJAX) feature which allows user to submit a form without any need of refreshing the page. There are three distinctive types of form submission in the DMS. These are URL submission, RGBA color submission, and general submission.

```
$stmt = $conn->prepare(
    "UPDATE css SET cssValue=? " .
    "WHERE cssName=?");
$stmt->bind_param("ss", $cssValue, $cssName);
```

The section above shows a predefined statement for the update function. The statement serves as a template which can be called multiple times. The statement shortens the code length substantially, because the repetitive queries are eliminated.

```
elseif (isset($_POST['cssValue4'])) {
    $url = $_POST["cssValue4"];
    $urlPrefix = "url(";
    $urlPostfix = ")";
    $cssName = "indexRow2BGImage";
    $cssValue = $urlPrefix . $url .
                $urlPostfix;
    $stmt->execute();
}
```

The section above shows an update process for a URL submission through the DMS. A specific function is created because some values should be able to show both an image through URL and a hex color input. The function adds a prefix and postfix strings which will add a URL property that is needed in order to generate the image on the web page.

```
function changeRowBorderColor() {
    var x = document.getElementById(
        "bottomColor").value;
    x = x.replace('#', '');
    r = parseInt(x.substring(0, 2), 16);
    g = parseInt(x.substring(2, 4), 16);
    b = parseInt(x.substring(4, 6), 16);
    a = document.getElementById(
        "borderOpacity").value;
    rgbacolor = 'rgba(' + r + ',' + g + ',' +
                + b + ',' + a / 100 + ')';
    document.getElementById(
        "borderRgbarResult").value =
        rgbacolor;
}
```

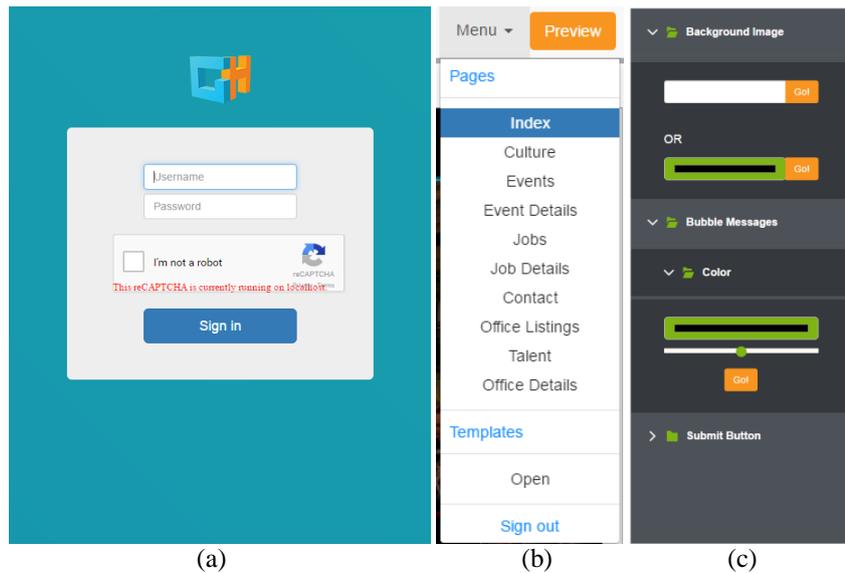
The second update function, which is the RGBA update, needs additional JavaScript function which will convert a hex color into an RGBA format. The reason why a hex color is submitted in the first place is because color input is used to submit a color option. Since the input is a defined property provided by HTML, therefore an additional procedure is needed to adjust the level of alpha, which is the opacity. After the function is executed, a regular submission can be performed.

```
elseif (isset($_POST['cssValue34'])) {
    $cssName = "indexButton";
    $cssValue = $_POST["cssValue34"];
    $stmt->execute();
}
```

The last type of submission is a general submission as shown in the section above. General in this term means that the value inserted in the DMS will be passed to the database as it is.

### 4.3. Templates

The feature allows user to load and save templates. The feature may come in useful when a major design change is needed for a certain situation, so that a user can save several design themes to be used in the future.



**Figure 5.** Login page (a), menu bar (b), and editor bar (c) of DMS

```

elseif (isset($_POST['cssSaveTemplate1'])) {
    unlink('../JSON/template1.json');
    $saveTemplate1 =
        fopen('../JSON/template1.json', 'w');
    fwrite($saveTemplate1,
        json_encode($cssValues));
    fclose($saveTemplate1);
}
    
```

The template is triggered when a button is clicked on the DMS. A template is saved as a JavaScript Object Notation (JSON) file to create a database as simple as possible. When saving values to a template, the previous saved template will be cleared to avoid conflict of values. Then, the next step will be opening the JSON file to write the array which contains database values inside the file. After the writing process is completed, the JSON file will be closed.

```

elseif (isset($_POST['cssLoadTemplate1'])) {
    $valueCount = count($cssValues);
    unset($cssValues);
    $loadTemplate1 =
        file_get_contents(
            '../JSON/template1.json');
    $template1 =
        json_decode($loadTemplate1, true);
    $cssValues = array();

    for ($x = 1; $x <= $valueCount; $x++) {
        $cssValues[$x] = $template1[$x];
        $cssValue = $template1[$x];
        $cssId = $x;
        $loadTemplate->execute();
    }
}
    
```

Loading a template is processed through the algorithm shown in the section above. The logic is similar with the save function. First, the array that defines the database values is cleared. After clearing the array, the JSON file is opened. The data inside the JSON file will be pushed to the empty array

accordingly through a loop. After populating the array, the predefined statement of updating will be called again and will go through a loop which will set data according to its CSS identity number.

### 5. Result

When opening the DMS, the user will be asked to enter their credentials before being redirected to the main page, as shown in Figure 5(a).

Once the user successfully enters his credential, the user will be navigated to the main page. There are three main components in the DMS, which are Editor Bar, Site Preview, and Menu Bar. This is the only page available in this editor since having a Single Page Application (SPA) is required to alleviate the learning curve of using this application. By having one source of truth, user will be more settled in using the features, since everything is there to be interacted with.

Figure 5(b) shows the menu bar. It is built by elements such as editable pages, templates, and the option to sign out from the application. If a page is selected, the site preview and the editor bar will be changed to the designated page and editing options.

Figure 5(c) is the editor bar. The bar provides editing options such as URL inputs, color inputs, and opacity slider. The bar will be changed according to the selected page.

The site preview is basically an iframe, which is an HTML element which gives the page an ability to show a page inside another page. The site preview's page will also change according to the selected page in the menu bar.

The application is regularly tested by stakeholders in the end of each sprint. The reason to do so is to adjust the current state of the application with the requirements of the stakeholders, lest there are any sudden changes throughout the implementation period. The latest version of the application is approved by the

stakeholders and it is a functional application that can be used anywhere to adjust design properties of the website.

## 6. Conclusion

After implementation, it can be inferred that:

- The application has the feature to alter CSS values by form submission through the DMS.
- Additional features that complement the DMS, such as templates and login page serve as valuable features for editing process.
- The implementation of a Single Page Application (SPA) to ease the usability is a success.

## References

1. GameHouse, *App Promotion, PC Publishing, Game Distribution from GameHouse*, retrieved from <https://partners.gamehouse.com>.
2. Microsoft Developer Network (MSDN), *UML Activity Diagram: Guidelines*, retrieved from <https://msdn.microsoft.com/en-us/library/dd409465.aspx>.
3. Coyier, C., *CSS Variables with PHP*, 2009, retrieved from <https://css-tricks.com/css-variables-with-php/>.