

Software development of Content Management System in Online-CADCOM platform

Ognyan Chikov¹

Abstract – In this paper the software development of Content Management System in Online-CADCOM platform is described. The two building subsystems – Content Management Application (CMA) and Content Delivery Application (CDA) and the used MVC approach are presented as well. Their role and place in the platform are given. Elements of code and "Dashboard" responsive view from smart device are presented as illustrations.

Keywords – Content Management System, MVC approach, Software development, Front-end, Back-end, E-content

I. INTRODUCTION

Online-CADCOM is an online assisted platform for computer-aided design in communications [1]. It is built of large number of instruments, resources and links to already studied and estimated tools and would be very difficult to manage all these materials without a reliable method. This method is by using separate platform that can manage all of the content which is available and accessible through Online-CADCOM – this platform is called Content Management System. Content Management Systems can be developed as using procedural or object-oriented programming and also using MVC Framework. Online-CADCOM's Content Management System (CMS) is developed using MVC Framework approach also and the advantages that this method has, makes the platform a reliable method for manage the content of large web platforms. Additional plugin – jQuery is used for event handling optimizations and speed up loading of first pages in the management system. The platform allows many users to efficiently share e-content such as audio files, video files, pictures, graphics, knowledge materials and content in the different categories as described in [1]. The Content Management System controls the access to this content in the repository as described in [2]. It is used to dynamically change the content of the Online-CADCOM without using complicated program languages and upload files every-time to the web content server through file transfer protocol (FTP), as well as central platform software for maintenance and user management. Users are authorized through user-level security and redirected to the appropriate "index" page of the platform where it is located the main navigation through different categories and functionalities. The platform uses a relational database to store the page content, metadata and other useful information assets that are

needed for the system. System administration is done through browser-based interface and do not require the use of "fat" client. The browser-based interface is "responsive" and appropriate for using with all Operational Systems (OS) and standard resolutions and provides scalable vision and structure representation of the content. Using Online-CADCOM's Content Management System provide some advantages of implement the platform. One of the advantages that this platform gives is that the cost of developing the content is minimal. This means that future programmers work could only be needed for developing new functionalities and not for uploading content or expanding it. Another advantage of the Content Management System is that it makes Online-CADCOM easy to use and support.

First challenges of using Content Management System in Online-CADCOM are presented, then two separated parts that build up the Content Management System are described and different main building script modules are presented. Also different code controllers are presented.

II. CONTENT MANAGEMENT IN ONLINE-CADCOM

Using Content Management System (CMS) in Online-CADCOM platform raises a lot of questions about this approach for managing the content. As described in [1], the Online-CADCOM platform will include a large number of user files that must be accessible 24/7. This means that the platform will have a lot of pages with different type of data [3]. The challenges for creating a Content Management Systems are related with the directory structure of the platform because of the large number of files. As described in [4] it is very important to properly scale the directory structure. As showed in [4] the non-CMS platform directory structures are ended with the name and type of file. The reason this is done is because there is no platform for managing the content and the user is directly connected with the desired file. If file is there, it is loaded, if not – there is an error. This approach is easy to implement and simple but not suitable for our case. That why as described in [4] in Online-CADCOM is used a directory structure approach that is implemented with CMS. Because the chosen approach is database-backed, it is not necessary to have a separate physical source file for every page. This approach solves the challenge related with the large number of e-content and materials in Online-CADCOM. Furthermore, the Online-CADCOM Content Management System does not write a physical file when a new page in the administration is done, it uses a "controller" to read from a database, based on what the URL was. This reduces the need for the CMS to have write-permissions for the publicly visible parts of the platform,

¹Ognyan Chikov is with the Faculty of Telecommunications at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: ognyan.chikov@tu-sofia.bg.

which increase the security as described in [5] and solve the security challenge.

Database structure is another challenge facing the Content Management System. For optimization purposes, Online-CADCOT should make as few database queries as possible. As described in [5] database of Content Management System is optimized using the following techniques:

- Proper indexing strategy
- Proper database schema
- Using external cache

As described in [2], the Content Management System is built of two separated parts that works interconnected:

1. Online-CADCOT Content Management Application (CMA) - it is a front-end user interface that allows all the users of the system to add, modify and remove content from a web site without the intervention of a programmer. This part of the platform is the part that users can see and operate with it. This is the front-end, the design of the visual part that contain all the categories, menus, buttons and forms for filling. Also this part includes all font, color and other visualization documents and scripts.

2. Online-CADCOT Content Delivery Application (CDA) - it's a powerful program script (backend) that compiles all the information that is provided from the CMA piece of program and use it to update the content of the website platform. Here work all the scripts on low login level. For example, the main CRUD (Create, Read, Update, Delete) methods and etc.

III. SOFTWARE DEVELOPMENT OF CONTENT MANAGEMENT APPLICATION (CMA)

The Content Management Application (CMA) is the front-end user interface of Content Management System (CMS). The idea of the Content Management Application is to take in hands the user interactions in the platform and later in the Content Delivery Application (CDA) - to perform this interaction. The front-end interface of the Content Management System is created with following rules as described in [6]:

- It is developed for multiple browser access – browsers are supposed to render webpages in the same way, but they don't and that is the reason why the code is developed to look correctly on multiple browsers
- The design is responsive – the CMS platform will be accessed from multiple and variety of devices – from smartphones and smart gadgets to notebook and desktop computers. Using CSS media queries, the management platform is with implemented responsive design.
- Colors are clean and simple – colors chosen for the design are simple and intuitive.
- Navigation is intuitive – Big problem for users operating with new platforms are to navigate for the first couple of times through the content. The CMS platform is made with big, simple and useful navigation menu and elements (buttons).

One of the most important elements of Content Management Application (CMA) is the “Dashboard”. The Dashboard is the first page that the user sees after successful login in the administration platform and gives immediately information about the productivity of Online-CADCOT. It is developed using the methods listed above for multiple browser access, responsiveness, simplicity, intuition navigation and user sight visibility. Figure 1 represents where the dashboard is located in the system architecture.

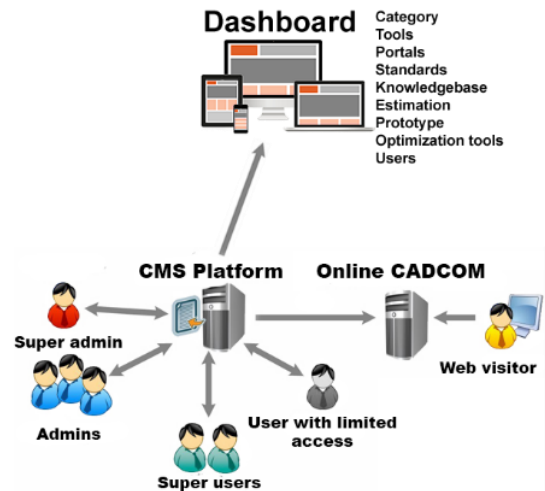


Fig. 1. Overview of development approach

As described in [3], Online-CADCOT has three main panels with content – Panel 1, Panel 2 and Panel 3. As Panel 1 and Panel 2 cover the Daniel-Gajsky Y model, these panels can contain more than one subcategory and in the same time these subcategories can contain more than one subcategories containing CAD tools. This statement makes no sense to hardcode the categories in the Panel 1 and Panel 2 and makes them static and impossible for changing. That's why the approach used in creating the Dashboard is to provide a dynamic method to add unlimited numbers of categories that contain subcategories and also unlimited number of tools that are stored in these different categories or/and subcategories. Dashboard manages all the related content in the platform and has additional menus:

- Portals,
- Standards,
- Knowledge Base,
- Estimation,
- Prototype,
- Optimization Tools,
- Users,
- Current User profile.

Figure 2 represents part of the responsive view of “Dashboard” loaded on a smart device.

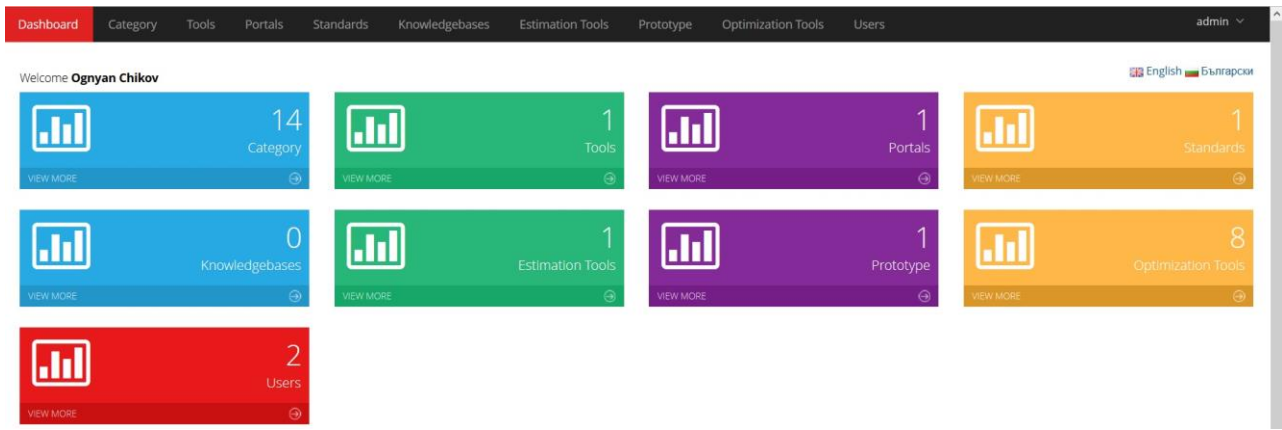


Fig. 2. “Dashboard” responsive view from smart device

IV. SOFTWARE DEVELOPMENT OF CONTENT DELIVERY APPLICATION (CDA)

Content Delivery Application is consisting of a large number of script files, compiled together which make possible all user interactions to take in use right after they were accomplished. The program code is called and changes are visible to the person who is editing the content and of course for the end user of Online-CADCOM. Used method and approach for CMS is highly complicated and reliable for future development and support. This is true because as described in [2] the approach that is used to build up both Online-CADCOM and CMS is with MVC Framework. The compiled platform is made of a hundreds of files, but one of the most important and major file that takes care for navigation through the “routes” and the “methods” of the platform is “routes.php” file [7]. Its content can be divided logically on two parts:

- Routes that are related with the CMS platform – these routes take care about the processes that are in the Content Management System (Admin routes)
- Routes that are related with the front-end part of Online-CADCOM – these routes take care about the processes that are in Online-CADCOM (Front-end routes)

Figure 3 represents how “routes.php” looks. File “routes.php” contains all the routes and controllers in the platform and this approach is covered in [7].

First step to reach the CMS platform is to request the URL “cadcom.demo/login”. When a user request the URL for the login screen, the browser send request for “cadcom.demo/login” to the file “routes.php”. Here if the “login” is not described, then the browser will return an error page. This is the reason why all the “URLs” are strictly described in the file showed on Figure 3 and they have a special function – to call a specific controller according to their functions.

```
Route::get('login','AdminController@showLogin');
Route::post('login', array('uses' => 'AdminController@doLogin'));
Route::get('logout', array('uses' => 'AdminController@doLogout'));
Route::get('registration', array('uses' => 'AdminController@showRegistration'));
Route::post('registration', array('uses' => 'AdminController@doRegistration'));
Route::get('forgot-password', array('uses' => 'AdminController@showForgotPassword'));
Route::post('forgot-password', array('uses' => 'AdminController@doForgotPassword'));

/*
|-----
| Front end - routes
|-----
*/

Route::get('/', 'FEHomeController@index');
Route::get('portals', 'FEPortalsController@index');
Route::get('portals/{id?}', 'FEPortalsController@show');
Route::get('standards', 'FEStandardsController@index');

Route::get('knowledgebase', 'FEKnowledgebaseController@index');
Route::get('ajax-subcat', 'FEKnowledgebaseController@getRecords');

Route::get('economical-estimation', 'FEEconomicalEstimationController@index');
Route::get('proto-type', 'FEPrototypeController@index');
Route::get('optimization-tools', 'FEOptimizationToolsController@index');
Route::get('contact', 'FEContactsController@index');
Route::post('contact', 'FEContactsController@store');

/*
|-----
| Admin routes
|-----
*/
Route::group(['before' => 'auth|admin'], function()
{
    Route::group(array('prefix' => 'admin'), function()
    {
        Route::get('dashboard', 'AdminDashboardController@getIndex');
        Route::resource('users', 'AdminUserController');
        Route::resource('category', 'AdminCategoryController');
        Route::resource('tools', 'AdminToolsController');
        Route::resource('portals', 'AdminPortalsController');
        Route::resource('standards', 'AdminStandardsController');
        Route::resource('knowledgebase', 'AdminKnowledgebaseController');
        Route::resource('economical-estimation', 'AdminEconomicalEstimationController');
        Route::resource('proto-type', 'AdminPrototypeController');
        Route::resource('optimization', 'AdminOptimizationToolsController');
    });
});
```

Fig. 3. Overview of development approach

With the given example above, when a user tries to reach the login page of CMS platform, the following route is checked and performed:

Route::get('login','AdminController@showLogin');

Route method that is used is “get”. The “get” method send the information together with the URL to the controller “AdminController” and calling function “showLogin”. In this controller is built-in the logic for user’s login. Below is shown how user’s input information is taken and used for the login function:

```
public function doLogin()
{
    $rules = array(
```

```

        'username' => 'required',
        'password' => 'required|min:4'
    );
    $validator = Validator::make(Input::all(),
$rules);
    if ($validator->fails()) {
        return Redirect::to('login')
            ->withErrors($validator)
            ->withInput(Input::except('password'));
    } else {

        $userdata = array(
            'username' => Input::get('username'),
            'password' => Input::get('password')
        );

        if (Auth::attempt($userdata)) {
            return Redirect::to('admin/dashboard');
        } else {
            return Redirect::to('login')-
->with('login_error',
trans('common.incorect_user_pass'))
            ->withInput();
        }
    }
}

```

“AdminController” is the most important controller in the platform. It contains the following methods in it doing these major functions:

- showLogin() –method which loads the “view” of the login form
- doLogin() – method which process the data from the login form
- doLogout() – method which destroys the current session of the user and log him out of the system
- showRegistrationForm() – method which loads user registration form
- doRegistration() – method that process the registration of the new user
- showForgotPassword() – method that loads a form for resetting the password
- doForgotPassword() – method that process the data and if it is valid send an email with a new password.

Second step to reach the CMS platform is to load the “Dashboard”. In the showed code above, if the variable “\$userdata” return true, then the URL is redirected to ‘admin/dashboard’ and again this URL is processed from the file “routes.php”. Again the requested URL is checked in it and it calls the line:

```
Route::get('dashboard','AdminDashboardController@getIndex');
```

In this controller the main methods that are used to perform showing and changing of information are:

- Index() - Display a listing of the resource.
- Create() - Show the form for creating a new resource.

- Store() - Store a newly created resource in storage.
- Show() - Display the specified resource.
- Edit() - Show the form for editing the specified resource.
- Update() - Update the specified resource in storage.
- Destroy() - Remove the specified resource from storage.

V. CONCLUSION

Online-CADCOM together with Content Management System forms a complicated software application that works on the web and that is reachable from everywhere and from any device with Internet connection. This makes the application with middle to high loading. The showed controllers and the methods that work inside are proof of quality and reliability of the code that ensures stable operation of the platform and avoids vulnerability. The used MVC Framework approach ensures easy maintenance of the code, good platform structure, lack of security bugs and easy traceability of platform logic. The described method for developing of scripts is reliable and highly effective for this type of web applications. The Content Management System is implemented for content development and task solution which will be presented in separate paper.

ACKNOWLEDGEMENT

This research is partly supported by Project №152PD0039-07 in Technical University – Sofia.

REFERENCES

- [1] G. Marinova, V. Guliashki, O. Chikov, “Concept of Online Assisted Platform for Technologies and Management in Communications – OPTIMEK”, Int.l Conference in Computer Science, Information System and Telecommunication, ICCSIST 2014, 7-9 November 2014, Durres, Albania, 2014, pp.55-62
- [2] O. Chikov, G. Marinova, MVC Framework Approach in the Online assisted platform form Computer-aided Design in Communications – Online-CADCOM, Computer and Communications Engineering, Submitted.
- [3] G. Marinova, O. Chikov, Methodology for tools integration in the Online Assisted Platform for Computer-aided design in communications, Proc. of papers of L International scientific conference on information, communication and energy system and technologies, ICEST’2015, 24-26 June 2015, Sofia, pp.31-36, ISBN:978-619-167-182-3
- [4] CMS Design Using PHP and jQuery, Kae Verens, Packt Publishing, 2010, ISBN: 978-1-849512-52-7
- [5] PHP5 CMS Framework Development, Martin Brampton, Packt Publishing, 2008, ISBN: 978-1-847193-57-7
- [6] V. Nikolov, D. Pilev, An Application of Cocoa Bindings and MVC Technology by Designing of a User Interface with Data Bases, Computer & Communications Engineering, 2/2013, pp.49-54, ISSN: 1314-2291
- [7] Web Content Management: System, Features and Best Practices – 1st Edition, Deane Barker, O’Reilly, 2015, ISBN: 978-1491908129