MVC Framework Approach in the Online assisted platform for Computer-aided Design in Communications – Online-CADCOM

Ognyan Chikov, Galia Marinova

***Abstract**: The paper presents the MVC Framework approach, implemented for the development of the platform Online-CADCOM. MVC programming approach is an architectural pattern that separates a web application into three main logical components: the Model, the View and the Controller, thus overcoming issues, rising from the complexity of web-based multitool design platforms. The architecture of the web application Online-CADCOM is described and it's illustrated by the "View" section from the MVC Framework model.*

***Key words:** Computer-aided design in communications, MVC Framework, Programming Layers, File Separation Content Management System*

1. INTRODUCTION

The platform Online-CADCOM, described in [1] manages links to online tools for computer-aided design of electronic and communication systems, which are previously studied, estimated and provided with characterization passports. The software requirements to the platform Online-CADCOM are described in [2]. The complexity of Online-CADCOM, its priority orientation forward open source and downloadable tools and resources, its concept as a platform of integration and exchange of resources and knowledge, represent a challenge for the software approach to be implemented. It must provide the ability for everyday calculations, as well as the coordination of different online resources and complex tasks solving. The "separation of concerns" which is achieved through the MVC Framework (Model, View, Controller) pattern, is considered as a suitable solution for software development of the platform. As noticed in [3,4], benefits from this technology are code synchronization between different components, ease of use, as well as software support for different models. The platform structure is set up, as well as the model for handling data representation and operation, the way data is viewed by users and the code that links the user to the system through events and requests. MVC pattern is widely used in online calculation tools [5]. It gives the developers a powerful instrument for creating resources to calculate values and in the same time to achieve the separation of project. Thus View and user events are passing to the Controller, which calls methods in the Model. With the alternative approach in a projects of this type of scope, all the code – views, functions, methods, program logic should be in one file and the maintenance of this tool would involve too much resources. Another fact which is taken into account, when choosing MVC Framework approach for the design and architecture of Online-CADCOM is its complexity and the scalability of its logic. Commonly used user platforms with different types of content, which are using the MVC programming approach, are: Platforms for web purchasing (eBay, Amazon, eMag); Social network platforms (Myspace, Facebook, Twitter, LinkedIn; Engineering web portals (Stack Overflow, LinkSprite), etc.

First the MVC Framework approach implemented in Online-CADCOM development is described and then the Model, View, Controller, and the Content Management System are presented.

2. MVC FRAMEWORK APPROACH IN ONLINE-CADCOM

The MVC Framework approach to design and build software platform Online-CADCOM consists in:

- Determining the architecture of Online-CADCOM and integrating components in MVC model and
- Setting up the MVC Framework for Online-CADCOM platform architecture.

2.1. Architecture of the Online-CADCOM platform and integration of components in the MVC Model

The Online CADCOM platform is interactive and user-friendly and it's separated on different user-interface components for easy access, collaboration and experience. This approach makes the platform rich of program components which have to be separated and in the same time to works globally. A lot of tools and resources have to find business logic place in the platform and its architecture should be able to take care of all components and in the same time all the components have to be separated. Figure 1 represents the MVC Framework architecture that is used to develop Online CADCOM software platform.

Reference [2] points the complexity of the web solution and the need of scalability. The architecture of the platform consists in 3 user panels which leads to additional resources. These additional resources could be any kind of content – other panels for navigations, as well as files and media content.

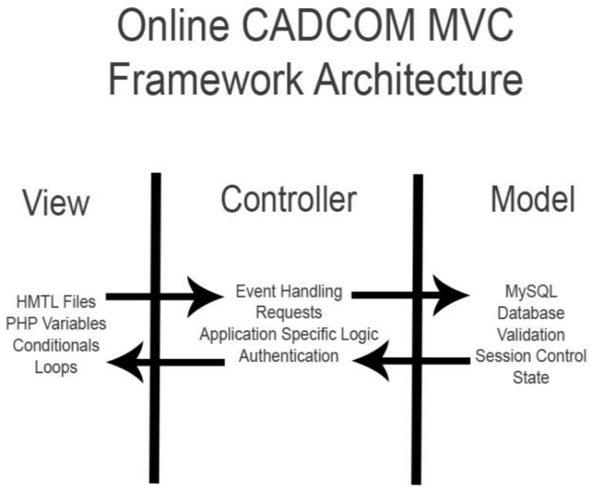


Fig. 1. *Online CADCOM Framework Architecture*

This part of the design representation corresponds to the View layer of the MVC model framework. This is the place where data requested from the Model layer is viewed and its final output is determined. In Online-CADCOM the view is the part of the system, where the HTML is generated and displayed, and the reactions from the users that interact with the Controller layer take part. The platform as shown in [1, 2] is rich of logics and functions. Here, the MVC framework Controller layer is entering in action. Controllers are containers for methods that contain the application logic of the web platform. The controller is containing the application logic that controls the Online-CADCOM Platform flow and it delivers the data from the database to the View, for proper formatting and displaying for users. The controller allows also some kind of group logic that applies configurations to a controller, which will affect all of the methods inside it. For example, one of the most frequently used controllers in a web project, is the user controller. Finally the MVC framework Model layer is the instrument that takes part of the architecture of the system, which manages all the tasks related to the data in the platform Online-CADCOM: validation, session control, state and database. The Model is responsible for the business logic and it reduces the complexity of the code by encapsulating methods to access the database and making reusable class library available. The code that is used to manipulate data in the database and to enforce business rules is also included in the Model layer. Online-CADCOM database using MVC framework, is a database, which is migration available, which keeps data of data structure and data itself. The system is using xml files to store data between migrations, making the process an easy task.

Table 1. Characterization of framework components in Online-CADCOM

FRAMEWORK VIEW	**VIEWS IN ONLINE-CADCOM**
HMTL Files PHP Variables Conditionals Loops	• Panels 1, 2, 3 (left Menu buttons), Header, Footer, Main Screen. • Resource Connections for scripts: • (JS validation of clients (jQuery form validation plugin)), • styles (CSS – style.css, custom.css bootstrap.css), • fonts (Arial, Times New Roman); • Deliver Videos, Music, Documents, etc.
FRAMEWORK CONTROLLER	**CONTROLLER IN ONLINE-CADCOM**
Event Handling Requests Application Specific Logic Authentication	• Render process of showing content. • Presentation of data that users are requested through different panels of the platform (panel 1, panel 2, panel 3) and different buttons – RF, PCB Design, Elements, etc. • Handle users request for specific URLs (panel 1, panel 2, panel 3) • Check users login status (What kind of users – super admins, admins, super users, users or guests. • Use model to gain access to specific data that users request – Knowledge Base, Manuals, Videos and other content. • Sends requested data to the VIEWs
FRAMEWORK MODEL	**MODEL IN ONLINE-CADCOM**
MySQL Database Validation Session Control State	• Tool passports content • CAD tools content • Knowledge Base • Manuals and Examples • Videos and other content • User Data tables with usernames, passwords and access rights

Table 1 indicates which part of the architecture of the platform is developed in the View framework model and what kind of platform components are included in VIEWs, which part is in the Controller layer of MVC framework and it shows how the specific logic of the platform is implemented. Components of the controller are also included. It presents which part of the architecture of the platform is in the Model layer and the components building this layer. As Table 1 shows, using MVC Framework pattern makes possible to orient and place platform components in the area of appurtenance of each of them.

2.2. Setting up the MVC Framework for Online-CADCOM platform architecture

The Online-CADCOM platform is set up using Laravel Framework, described in [6], which offers true code modularity and easy change and extend caching, session, database and authentication.

The set up process used, is through “Composer Create-Project”. To use this method of installation, the “composer” was installed first. This method is working through the console application and set up process is done using the command:

```
composer create-project laravel/laravel “online-cad” 4.2 --prefer-dist
```

The configuration of the project is done through all the configuration files, stored in the “config” directory. The configuration of the framework for the web platform is done through basic configuration of several points:

- Directory permissions;
- Application Key;
- Additional Configuration of Cache, Database and Session;
- Pretty URLs.

The database of Online-CADCOM was set up using MySQL because this type of database is easy to maintain and in the same time it provides high availability and security, as pointed in Reference [9].
An important part of working with any framework application is to set up a virtual host. Online-CADCOM is using the virtual host: “online-cadcom.demo”. This is made because of later easy implementation of the URLs with the online web servers and hosting configurations. Next, the source code shown, is from both files “hosts” and “httpd-vhosts.conf”. The first one is located in C:\Windows\System32\drivers\etc\hosts and in it we write the localhost host as a name. And the second one is located the apache configuration folder.

From “hosts”:

```
127.0.0.1 localhost
127.0.0.1 online-cadcom.demo
```

From “httpd_vhosts.conf”

```
#online-cadcom.demo
<VirtualHost *:80>
DocumentRoot “D:\wamp\www\online-cadcom\public”
ServerName online-cadcom.demo
</VirtualHost>
```

Lines above are related to creating a virtual host, which is used for development of the software platform.
Model, View and Controller sections are automatically set up within the framework installations. As shown on Fig. 2 – they are represented in different folders. These folders contains sample files, so newer users can orientate in the structure of the framework oriented platforms.

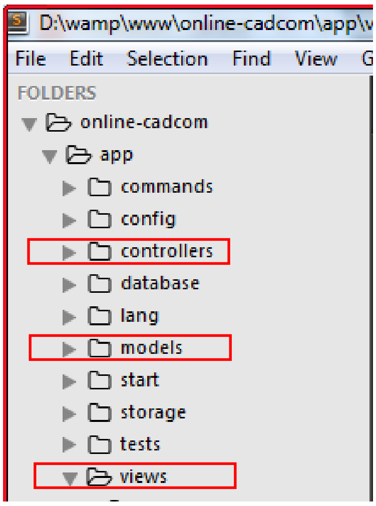


Fig.2. *Models, Views and Controllers folders*

The MVC data models are represented in app/view, app/controllers and app/models. This separation makes the code of the online platform cleaner. These are the folders where all files of different parts of the platform are saved. This approach is used to achieve clear and highly understandable logic representation of the web platform.

3. APPROACH OF MODEL, VIEW, AND CONTROLLER DEVELOPMENT

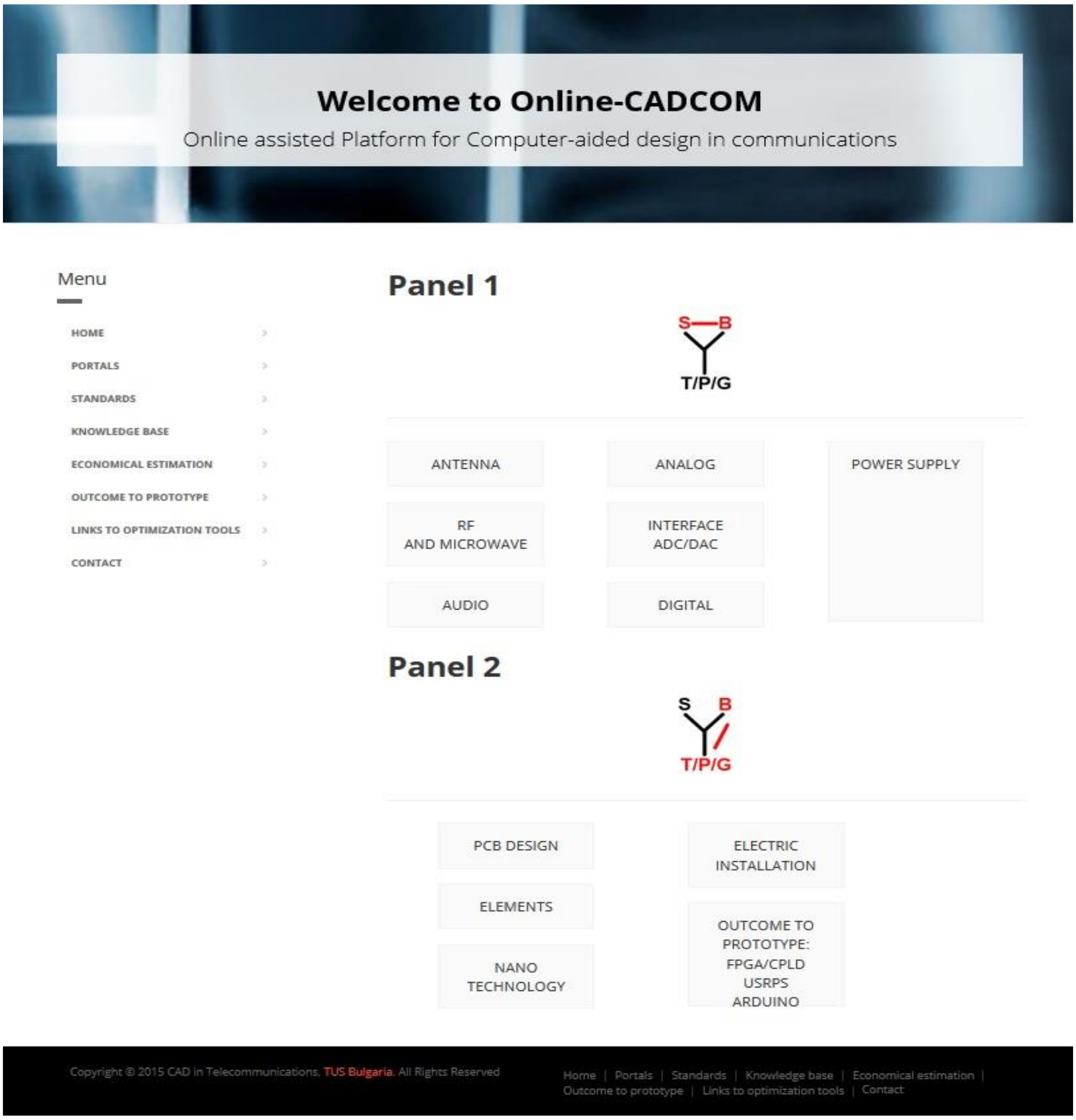
Using the MVC framework approach for web developing of Online-CADCOM, requires the development of all the components simultaneously, because technically it’s not possible to do the controller without the model and vise versa [3, 9]. The View could exist without the other two, as a simple HTML and CSS files.

First of all, the View layer is presented. The html files are developed right in the framework development environment. The approach, used is to separate different parts of views:

- The header of the platform is static, so it is presented in separated file.
- The content of the platform is dynamically changeable, so it is separated in different file using different connections with the controller.
- The footer of the platform is static, so it is presented in separate file. All the CSS files are separated and grouped, as the jobs, they are related to.
- All the script files (javascript, jquery, etc.) are separated and grouped, as the jobs, they are related to.

This approach is used to ease and help View changes to be performed [8]. The code is separated in different files, which are representing different part of the view sections – “default.blade.php” takes care of the platform header, “home.blade.php” takes care of the representation of the content of the platform.

- Next figures show the explained method of View components separation. Figure 3 shows the main screen of “index.php” file, loaded in a browser. Here the simple user is not affected from the separation of the different parts of the views and the web design is shown properly.



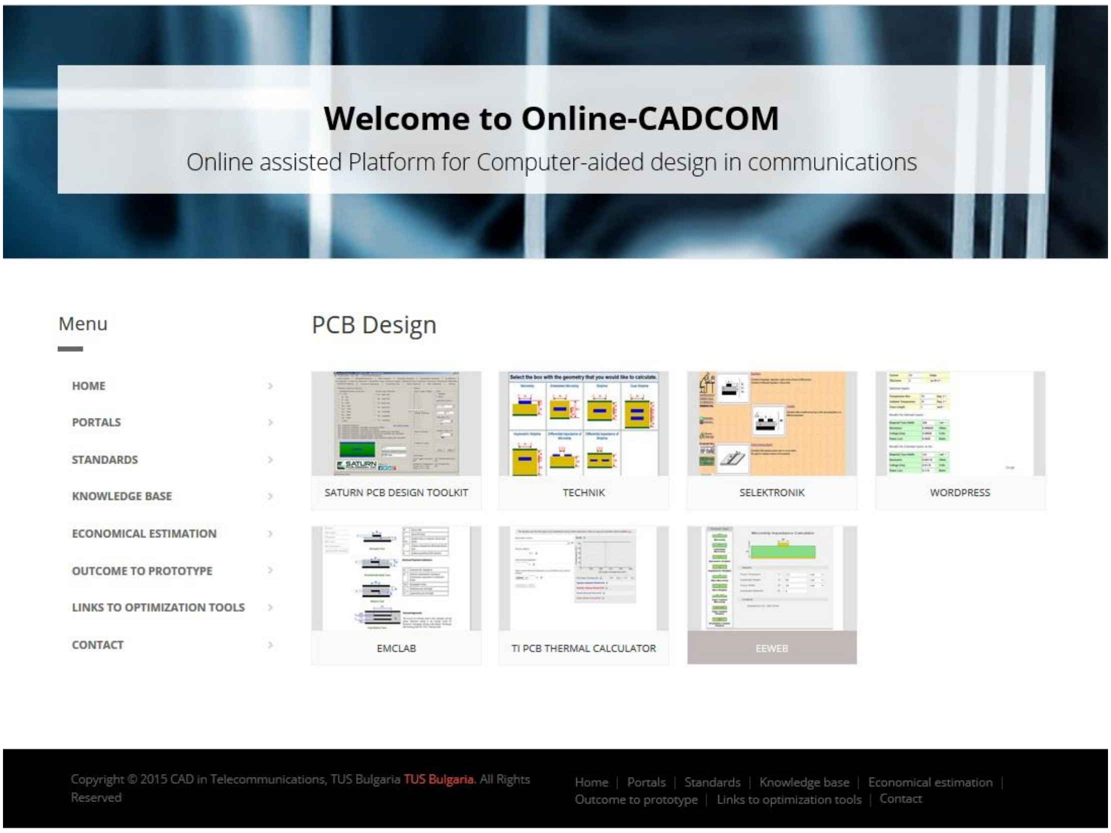
Welcome to Online-CADCOM

Online assisted Platform for Computer-aided design in communications

Copyright © 2015 CAD in Telecommunications, TUS Bulgaria. All Rights Reserved

Fig.3. *View from the main screen of Online CADCOM*

.



Welcome to Online-CADCOM

Online assisted Platform for Computer-aided design in communications

Copyright © 2015 CAD in Telecommunications, TUS Bulgaria TUS Bulgaria. All Rights Reserved

Menu
PCB Design
HOME
PORTALS
STANDARDS
KNOWLEDGE BASE
ECONOMICAL ESTIMATION
OUTCOME TO PROTOTYPE
LINKS TO OPTIMIZATION TOOLS
CONTACT
SATURN PCB DESIGN TOOLKIT
TECHNIK
SELEKTRONIK
WORDPRESS
EMCLAB
TI PCB THERMAL CALCULATOR
EEWEB
Home | Portals | Standards | Knowledge base | Economical estimation | Outcome to prototype | Links to optimization tools | Contact

Fig. 4. *Submenu content in Online-CADCOM*

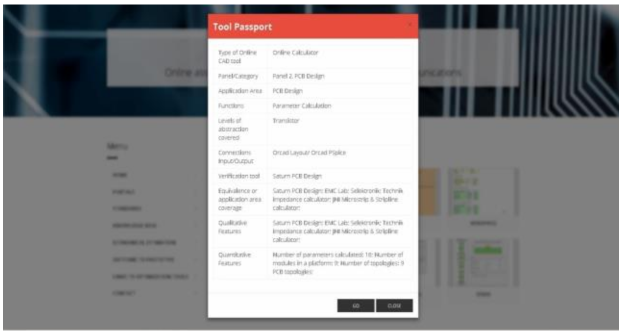
This approach gives us the ability to solve the complex tasks, related to the platform logic – it is high scalable and logic is not fixed. Figure 4 shows a subpanel, which is shown when button “PCB Design” from “Panel 2” is pressed. View section is not accomplished entirely with the main screen of the web platform. As shown in Reference [2], Online CADCOM will have different type of resources, with different types of representations, different types of logic level and access. This requires further development of the View framework layer. Each “panel” button is leading to another subpanel of buttons. This structure is not fixed. It is dynamically changeable, thanks to the modern approach used. Prototyping and optimizations tools are shown, as their data and connections are controlled from the Controller layer and represented from the Model layer database.

All online resources for prototyping and optimizations have unified “Tool Passport”. This item can be included at different logic levels of the platform. MVC framework approach gives the opportunity not to hardcode the logic level of architecture representation, but to use dynamically content scripts as points [11]. Source code which present the tool passport is as follows:

```
<!-- Modal -->
<div class="modal fade" id="myModal" tabindex="-1"
role="dialog" aria-labelledby="myModalLabel" aria-
hidden="true">
 <div class="modal-dialog">
  <div class="modal-content">
   <div class="modal-header">
    <button type="button" class="close" data-
dismiss="modal"><span aria-
hidden="true">&times;</span><span class="sr-
only">Close</span></button>
    <h4 class="modal-title" id="myModalLabel">Tool
Passport</h4>
   </div>
   <div class="modal-body">
     <table class="table table-bordered">
<tbody>
       <tr>
        <td>Type of Online CAD tool</td>
        <td>{{ $calcularotInfo->type }}</td>
       </tr>
       <tr>
        <td>Panel/Category</td>
        <td>{{ $calcularotInfo->category }}</td>
       </tr>
       <tr>
        <td>Application Area</td>
        <td>{{ $calcularotInfo->applicationArea }}</td>
       </tr>
       <tr>
        <td>Functions</td>
        <td>{{ $calcularotInfo->functions }}</td>
       </tr>
       <tr>
        <td>Levels of abstraction covered</td>
        <td>{{ $calcularotInfo-> }}</td>
       </tr>
       <tr>
        <td>Connections Input/Output</td>
        <td>{{ $calcularotInfo->connections }}</td>
       </tr>
       <tr>
        <td>Verification tool</td>
        <td>{{ $calcularotInfo->verificationTool }}</td>
       </tr>
       <tr>
        <td>Equivalence or application area
coverage</td>
        <td>{{ $calcularotInfo-
>appAreaCoverage}}</td>
       </tr>
       <tr>
        <td>Quantitative Features</td>
        <td>{{ $calcularotInfo->quantitativeFeatures
}}</td>
       </tr>
      </tbody>
     </table>
   </div>
   <div class="modal-footer">
     <a
href="http://www.saturnpcb.com/pcb_toolkit.htm"><b
utton type ="button" class="btn btn-sm btn-
dark">Go</button></a>
    <button type="button" class="btn btn-sm btn-dark"
data-dismiss="modal">Close</button>

   </div>
  </div>
 </div>
</div>
```

As shown on Fig. 5, "Tool Passport" is designed to be a pop-up window using the libraries of jQuery JavaScript library.


Tool Passport

Type of Online CAD tool	Online Calculator
Panel/Category	Panel 2. PCB Design
Application Area	PCB Design
Functions	Parameter Calculation
Levels of abstraction covered	Transistor
Connections Input/Output	Orcad Layout/ Orcad PSpice
Verification tool	Saturn PCB Design
Equivalence or application area coverage	Saturn PCB Design; EMC Lab; Selektronik; Technik Impedance calculator; JNI Microstrip & Stripline calculator;
Qualitative Features	Saturn PCB Design; EMC Lab; Selektronik; Technik Impedance calculator; JNI Microstrip & Stripline calculator;
Quantitative Features	Number of parameters calculated: 18; Number of modules in a platform: 9; Number of topologies: 9 PCB topologies;

Fig.5. *"Tool Passport" View*

The content of each "Tool Passport" is maintained through the Controller and it's represented from the Model information which is in the MySQL database.
The Controller layer is the layer, which makes the connection between the View that screens are shown above and the Model layer – shown later in the paper.
Part of Online-CADCOM Controller is presented with the code bellow located in file "HomeController.php":

```
 <?php
class HomeController extends BaseController {
   public function index() {
      $calculatorType = Input::get('calculator_type');
      $calcularotInfo = DB::table('calculator')
         ->join('calculator_details', 'calculator.id', '=',
'calcularor_details.id')
         ->where('calculator_type', '=',
$calculatorType)
         ->first();
      return
View::make('front_end/solutions.solution_template')
         ->with('calcularotInfo' , $calcularotInfo);    }
```

This code includes method "index()", which takes the user's choice of interaction and pulls a data from the database based upon user's choice. The value of user's choice is stored in variable "$calculatorType". In the same time "$calculatorInfo" stores the information about the tool, which is taken from the database upon the user's choice. All the information from the variables is passed to the View.
The Model layer takes care of the user information and of all data in the platform The CRUD methods for user records have to be implemented at the Model layer. Code that is making route declaration is bellow, following the explanation in [10]:

```
Route::get(ochikov, function() {  $user = new User;
  $user->email = "ochikov@gmail.com ";  $user-
>real_name = "Ognyan_Chikov";  $user->password
= "qwerty123";
  $user->save();
  return "The ochikov user has been saved to the
database."; });
```

As this example cares only of adding new users to the system. The Code bellow shows the Model realization:

```
<?php
class Calculator extends \Eloquent {
   protected $table = 'calculator';
}
<?php
class CalculatorDetails extends \Eloquent {
   protected $table = 'calculator_details';
}
return View::make('front_end/online-cadcom')
```

4. CONTENT MANAGEMENT SYSTEM

The Online-CADCOM is using Content Management System (CMS) to change dynamically the content and its representations. Without this important part of the platform, it would be difficult for managing. Development of the CMS platform went further. The web platform is used, not only for dynamically changing of content, but also for maintenance from a central interface [3]. Figure 6 shows the method which is used in this particular platform.

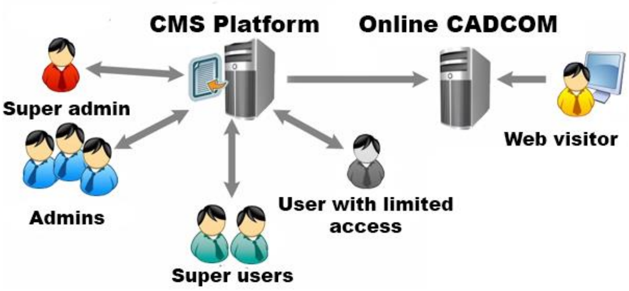


Fig.6. *Content Management System architecture of Online CADCOM platform*

This approach is used as pointed in Reference [7] for the following reasons:

- It eases storing and organizing files and folders and provides version-controlled access to their data.
- It's a powerful method for controlling centrally a lot of web platforms.
- It's the modern way for providing access to people who are not programmers, to modify the content of web platforms. It is made for website administrators.

The CMS in Online-CADCOM platform is separated in two parts:

- Online-CADCOM 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. Different levels of access are provided and there are hierarchy level of users – super admin, admins, super users, users with limited access.
- The Online-CADCOM Content Display 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.

5. CONCLUSION

The paper defines the main steps and methodology of using MVC Framework approach in the architecture and the design of the platform Online-CADCOM. It illustrates some advantages of using the chosen method of development. The illustrations, provided in the paper, confirm the suitability of this approach for the development of Online-CADCOM. The approach implemented might be used for other online CAD/CAE platforms. Additional results from particular layers development, screens, logic and code structure will be presented in future works.

Acknowledgment

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

REFERENCES:

[1] 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

[2] 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

[3] 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

[4] T. Dey, A Comparative Analysis on Modeling and Implementing with MVC Architecture, Int. Conference on Web Services Computing (ICWSC) 2011, pp.44-49

[5] Simple Example of MVC (Model View Controller) Design Pattern for Abstraction, 8 April 2008, http://www.codeproject.com/Articles/25057/Simple-Example-of-MVC-Model-View-Controller-Design

[6] Laravel, "Installation and first time use", 2015, http://laravel.com/docs/5.1/installation

[7] *P. Boag,*"10 Things To Consider When Choosing The Perfect CMS"*. SMASHING MAGAZINE,* http://www.smashingmagazine.com/2009/03/10-things-to-consider-when-choosing-the-perfect-cms/

[8] MVC Framework architecture and pattern, 2013, http://www.sitepoint.com/the-mvc-pattern-and-php-1/

[9] MySQL "What's new in MySQL 5.7", 2015, https://www.mysql.com/why-mysql/white-papers/whats-new-mysql-5-7/

[10] C. Pitt "Pro PHP MVC", Apress, 2012

[11] K. McArthur "Pro PHP Patterns, Frameworks, Testing and More", Apress, 2008

About the authors:

Ognyan Chikov, Ph.D. student, Faculty of Telecommunications, Technical University-Sofia, e-mail: ognyan.chikov@tu-sofia.bg

Galia Marinova, Associate professor, Ph.D. Faculty of Telecommunications, Technical University-Sofia, e-mail: gim@tu-sofia.bg