AN EMPIRICAL STUDY OF THE EVOLUTION OF PHP MVC FRAMEWORK

RashidahF.Olanrewaju[1], Thouhedul Islam[1] and Nor'ashikin Bte. Ali[2],
[1]Department of Electrical and Computer, Kulliyyah of Engineering
International Islamic University Malaysia
P.O Box 10, 50728 Kuala Lumpur, Malaysia
[2]College Of Information Technology, Universiti Tenaga Nasional Malaysia.
[1]frashidah@yahoo.com

Abstract:

Commercial, social and educational importance of web technology has tremendously increased research activities in web programming/scripting. Several methods for writing PHP codes such as Object Oriented Programming (OOP), Procedural PHP coding and Model View Controller (MVC) pattern have been proposed. Model View Controller (MVC) which is one of the most powerful method for developing PHP application has many variant such laravel, Symfony, CodeIgniter, CakePHP etc. However, selection of best MVC framework among the variants is of concern to the programmers as well as project managers, especially when managing big applications. Hence,performance evaluation criterions are required. This paper discusses the MVC based most famous PHP frameworks, evaluate their performance and it was found that Laravel outperforms other MVC framework, hence laravel is proposed as the most suitable PHP framework for future web technology.

Keywords: MVC, laravel, PHP Framework, cakePHP, CodeIgniter, Symfony

1. Introduction:

The rapid development of internet for web based application indicates a higher demand of reliability, scalability, security and maintainability of coding methodology. PHP, a scripting tool for web that enable dynamic interactive web development suchintuitive, compiled fast, cross platform, open source, flexibility as well as required minimal setup. This became one of the important web development language thus, PHP is one of the most powerful programming language in the web world. Several developers choose to deploy application based on PHP putting all the issues such as data access, business logic, and data representation layer together.This in turn createdevelopment problems especially for big projects. To solve this problem, MVC design pattern brings an effective ways to separate code in layers from each other based on each layer activities.

MVC design pattern is a proven effective way to develop application such as CakePHP, Codeigniter, Laravel, Symfony. The main methods of MVC to spilt an application into separate layer that can work separately and produce same result. The advantage of using MVC pattern are-

- Standard, consistency and predictability
- Software components or building-blocks so that developers can share and reuse code

- A model or standard architecture that allows easy visualization of how the entire system works
- Reusable and thoroughly tested code in the libraries, classes and functions.
- Well-structured code using architectural pattern.
- Security, interoperability and Maintenance.

Although MVC based framework (cakePHP, laravel, codigniter) has number of advantages,however, selecting of best PHP framework is still a concern. This is because all of the framework does not cover all aspect of web applications. This study evaluate most famous PHP frameworks based on MVC design model and it performance as well as proposed the best efficient PHP MVC framework for future web development.

2. Materials and Methods:

The operation of Model View Controller (MVC) method is to spilt or separate the different parts of code into layers such as view, data access, controllinguser's requests and forward request to relevant layers

The MVC pattern's title is a collation of three core parts: Model, View, and Controller. A visual representation of a complete and correct MVC pattern looks like the following in Figure 1.

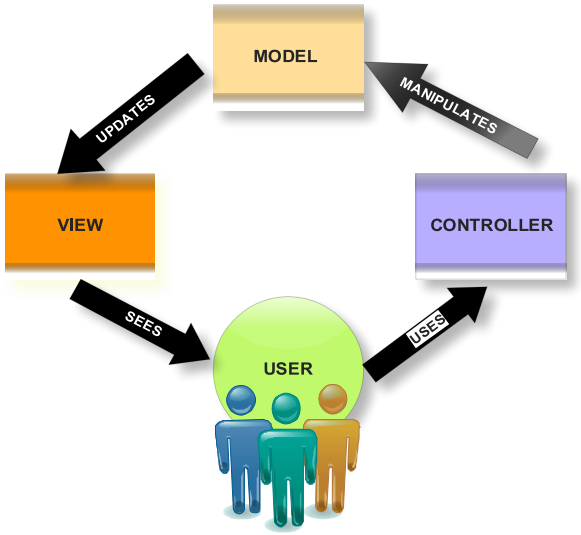

Fig 1: MVC Mechanism

Figure 1 shows the pattern and interaction with the user and the application itself. It is a single flow layout of data, how it's passed between each component, and finally how the relationship between each component works.

a. Model

The Model is the name given to the permanent storage of the data used in the overall design. It must allow access for the data to be viewed, or collected and written to, and is the bridge between the View component and the Controller component in the overall pattern.
One important aspect of the Model is that it's technically "blind" – by this, the model has no connection or knowledge of what happens to the data when it is passed to the View or Controller components. It neither calls nor seeks a response from the other parts of the component; its main purpose is to process data into its permanent storage, seek and prepare data to be passed along to the other parts.
The Model cannot simply be assumed as a database toolkit only, or a gateway to another system which handles the data process. The Model represents a gatekeeper to the data itself, asking no questions but accept all requests which comes its way. Often this most complex part of the MVC system, the Model component is also the pinnacle of the whole system since without it there will be no connection between the Controller and the View.

b. View

The View is a module where data, requested from the Model is viewed and its final output is determined. Traditionally in web application use MVC for development, the View is the part of the system where the HTML is generated and displayed. The View also ignites reactions from the user, who then goes on to interact with the Controller. The basic example of this is a button generated by theView, which a user clicks and triggers an action in the Controller.

Misconceptions about View

There are some misconceptions held about View components, particularly by web developers using the MVC pattern to build their application. For example, many mistake the View as having no connection whatsoever to the Model and that all of the data displayed by the View is passed from the Controller. In reality, this flow disregards the theory behind the MVC pattern completely. Fabio Cevasco's article The CakePHP Framework: Your First Bite [9]demonstrates this confused approach to MVC in the CakePHP framework.
In order to correctly apply the MVC architecture, there must be no interaction between models and views: all the logic is handled by controllers.Furthermore, the description of Views as a template file is inaccurate. The View is really much more than just a template, themodern MVC inspired frameworks have bastardised the view almost to the point that no one really cares whether or not a framework actually adheres to the correct MVC pattern or not.It's also important to mention that the View part is never given data by the Controller. There is no direct relationship between the View and the Controller without the Model in between them.

c. Controller

The third component of the triad is the Controller. Its job is to handle data the user submits as well as update the Model accordingly. The Controller can be summed up as a collector of information, which then passes it on to the Model to be organized for storage, and does not contain any logic other than collecting user input. The Controller is also only connected to a single View and to a single Model, making it a one way data flow system, with handshakes and signoffs at each point of data exchange. Controller is only given tasks to perform when the user interacts with the View first, and that each Controller function is a trigger, set off by

the user's interaction with the View. The most common mistake made by developers is confusing the Controller as a gateway, and ultimately assigning it functions and responsibilities that the View should do (this is normally a result of the same developer confusing the View component as a template). Additionally, it's a common mistake to assign the Controller functions that gives it the sole responsibility of crunching, passing, and processing data from the Model to the View. Nonetheless, the MVC pattern relationship should be kept between the Model and the View.

MVC Design Frameworks

a. Codeigniter:

CodeIgniter is an open source rapid development web application framework, for use in building dynamic websites with PHP. Its goal is to enable to develop projects much faster than writing code from scratch, by providing a rich set of libraries for commonly needed tasks, as well as a simple interface and logical structure to access these libraries. The first public version of CodeIgniter was released on February 28, 2006, and the latest stable version 2.1.4 was released July 8, 2013 [22]

CodeIgniter is loosely based on the popular Model-View-Controller development pattern. While view and controller classes are a necessary part of development under CodeIgniter, models are optional. CodeIgniter is most often noted for its speed when compared to other PHP frameworks.

b. CakePHP:

CakePHP is an open source web application framework. It follows the Model-View-Controller (MVC) approach and is written in PHP, modeled after the concepts of Ruby on Rails, and distributed under the MIT License.[19]
CakePHP uses well-known software engineering concepts and software design patterns, as Convention over configuration, Model-View-Controller,Active Record, Association Data Mapping, and Front Controller.

c. Symfony:

Symfony is a PHP web application framework for MVC applications. Symfony is free software and released under the MIT license. The symfony-project.com website was launched on October 18, 2005.[23]

d. Laravel:

Laravel is a free, open source PHP web application framework, designed for the development of MVC web applications. Laravel is released under the MIT license, with its source code hosted on GitHub.

The key design points of laravel are-

- Bundles provide Laravel with a modular packaging system, and numerous bundled features are already available for easy addition to applications.
- Eloquent ORM is an advanced PHP implementation of the active record pattern, providing internal methods for enforcing constraints to the relationships between database objects.
- Application logic is part of developed applications, either by using controllers, or as part of route declarations. Syntax used for definitions is similar to the one used by Sinatra framework.
- Reverse routing defines a relationship between links and routes, making it possible for later changes to routes to be automatically propagated into relevant links. When links are created by using names of existing routes, appropriate URIs are automatically created by Laravel.
- Restful controllers provide an optional way for separating the logic behind serving HTTP GET and POST requests.
- Class auto loading provides automated loading of PHP classes, without the need for manual maintenance of inclusion paths. On-demand loading prevents loading of unnecessary components; loaded are only those components which are actually used.
- View composers are logical code units that can be executed when a view is loaded.
- Migrations provide a version control system for database schemas, making it possible to associate changes in the application's code base and required changes in the database layout, easing deployment and updating of applications.
- Unit testing plays an important role in Laravel, which itself has a large number of tests for detecting and preventing regressions. Unit tests can be run through the artisan command-line utility.
- Automatic pagination simplifies the task of implementing pagination, replacing the usual manual implementation approaches with automated methods integrated into Laravel.[24]

3. Benchmarking

To evaluate the performance of four mentioned PHP framework; CodeIgniter (CI), Symfony, CakePHP and Laravel. The best way to do the benchmarkingis applying by several criteria such as request per second, system load average, memory usage, number of function calls and number of files required on each of the MVC. To evaluate these four framework, web design which contains “hello word” was carried out on apache (ab –c 200 –n 50000), below are results of each evaluation criteria.

a. Request per Second:

This benchmarking is based on apache (ab –c 200 –n 50000). Figure 2 shows the performance comparison among four MVC: CI, cakePHP symphony and laravel. It can be seen that Laravel outperforms other MVC in terms of request person. It was able to handle 3000 request per second compare to others. In this case, bigger output indicates perfect result hence, it denotes best performance.

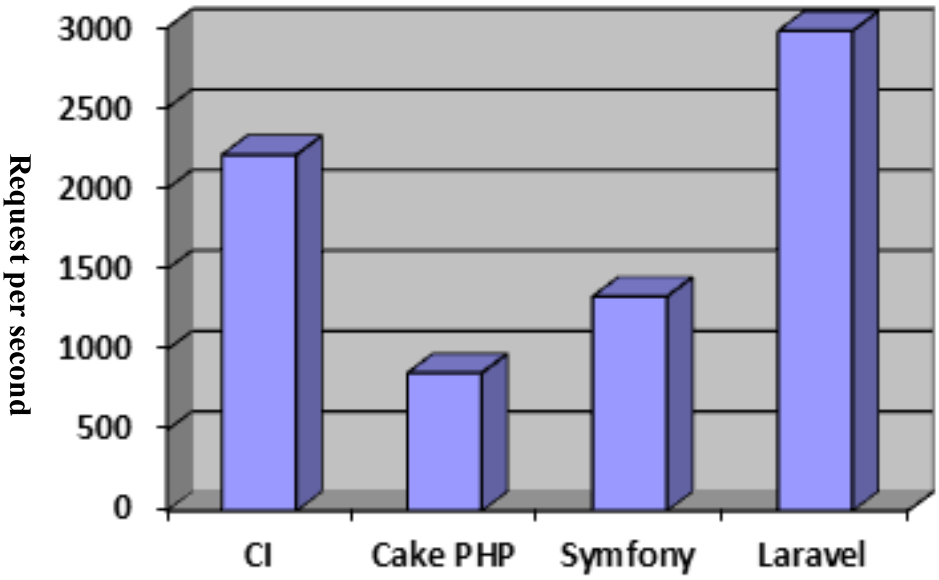

Figure 2: Request per second among PHP framework.

b. System Load Average

System Load Average in 1 minute when Apache Benchmark is complete, the smaller is better in the condition of ab –c 200 –n 50000).**Figure 3 show the comparison of the four MVC in terms of system load within one minute.** In this graph, laravel contain lowest times (.98) where cakephp contain maximum load time 5.1 per minute to load system. Based on average time, lowest average time is better to run MVC application.

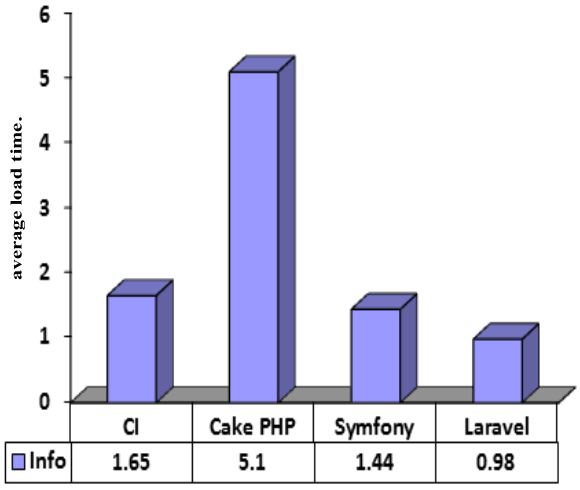

Figure 3: System average load time.

c. Memory usage

This benchmarking checks how much memory is used in a one word 'hello world' page display. The smaller number of KB is better is the memory usage. Figure 4 indicated that Laravel is about 518KB compare to CI which is which 725.Kb follow by symphony with memory usage of about 1711 then CakePHP with 2824KB. Hence, it can be concluded that Laravel is memory usage efficient.

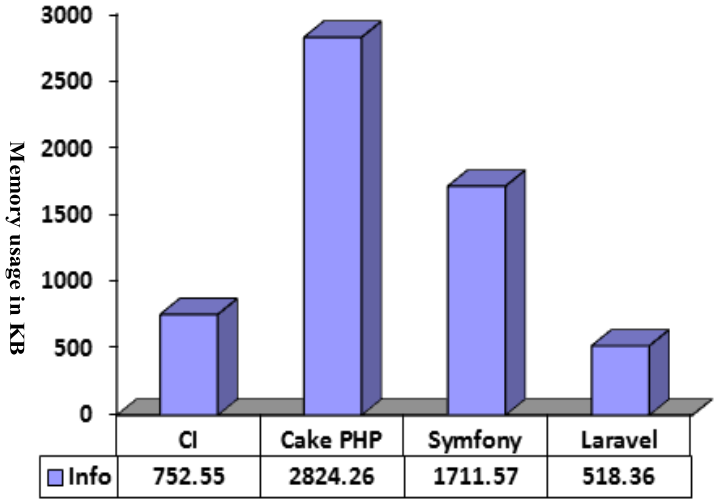

Figure 4: Memory usage.

d. Response Time

The time of page request to response from framework is one of the most important criteria to evaluate MVC performance. It is calculated by millisecond. The lower number of millisecond of an MVC calculated the better performance. Figure 5 depicts the result of all the four MVC used. Among all, laravel came out to be with the least response time, 4.46milliseconds compare to CI with 7.2 followed by symphony with 12 then CakePHP with about 14milliseconds.

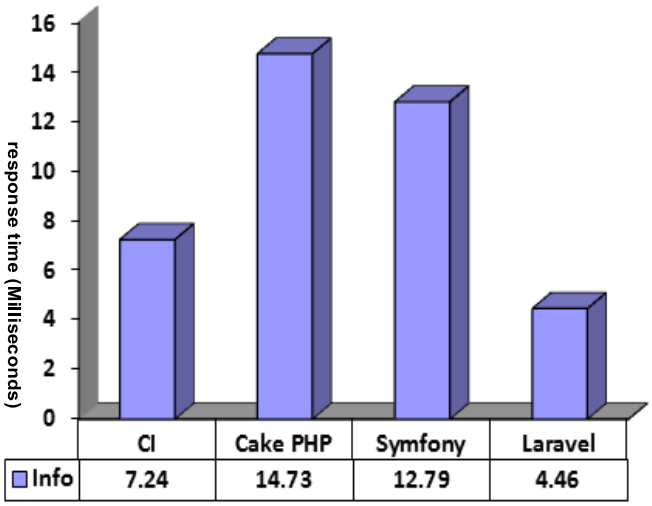

Figure 5: Response time for various MVC

e. Number of function calls

This test checks the how many functions are calling for one "hello world" pages in terms of Facebook XHProf [20]. Here smaller number of functions is most effective for php framework evaluation. It can be seen from Figure 7, Laravel outperforms other MVC framework with the minimal number of function calls with 238 calls compare to CakePHP 834 and others.

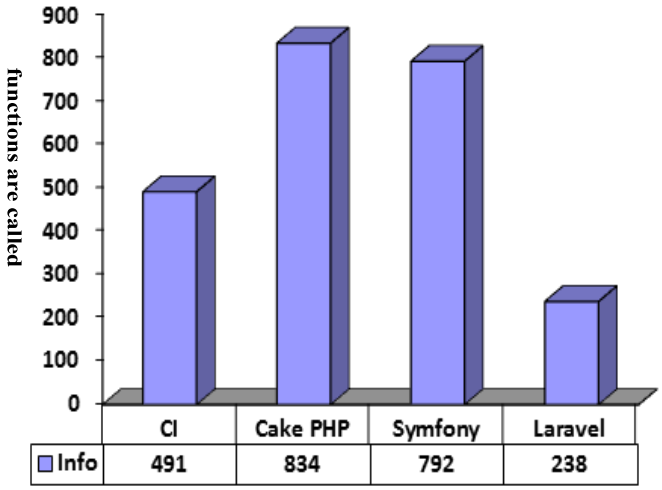

Figure 6: Numbers of functions called.

f. Number of Files

The number of files included or required in one 'hello world' page. Less amount of required filed represent that kind of framework will be loaded first in-terms of file running. Smaller numbers of required files are highly appreciable. As shown in Figure 7, CI comes up with the least file of 22, then symphony with 15 files followed by laravel with 26, while CakaPHP loaded 37 files

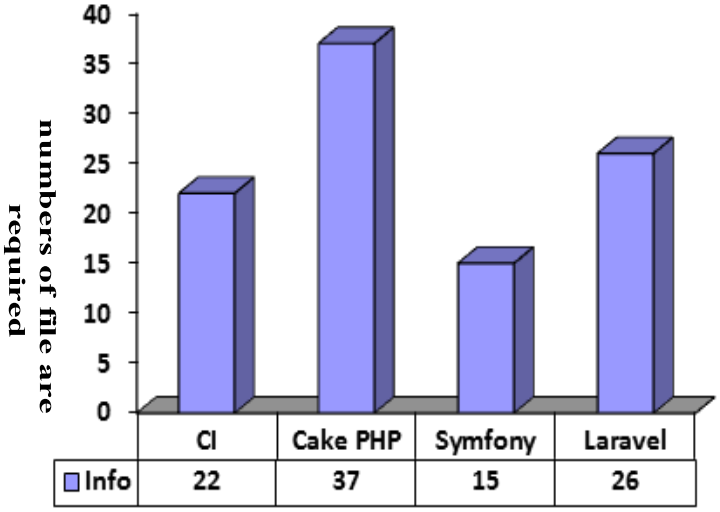

Figure 7: Numbers of file are required by various MVC.

3. Results and Discussion:

Based on different core criteria of PHP framework suchas benchmark, pattern, database access, field database, session, cache and library, this research compare among four top PHP framework (laravel, cakePHP, CodeIgniter & Symfony) and their performance. Table 1 shows comparisons the frameworks in terms of facilities.

Table 1: Comparism among four MVC framework

Specification	CakePHP	CodeIgniter	Laravel	Symfony
MVC	✓	✓	✓	✓
DB	✓	✓	✓	✓
Fielddb	✗	✗	✓	UK
Auth	✓	✓	✓	✓
Validate	✓	✓	✓	✓
Session	✓	✓	✓	✓
Cache	✓	✗	✓	✓
Ajax	✓	✓	✓	UK
Db	✗	✗	Eloquent	✓
MVC type	PMVC	UMVC	HMVC	UMVC
MVC db	AR	AR	ORM	ORM
Upload	✓	✓	✓	✓
Form	Objects	Procedural	Objects	Objects
Xml	✓	✓	✓	✓
library	✗	✗	✗	UK

From the table 1, it is assumable that, laravel has advantage over other MVC PHP framework.

The four frameworks was also measuredbased on current trends for the future PHP framework that will lead the next generation of web. Fig 2 shows the details.

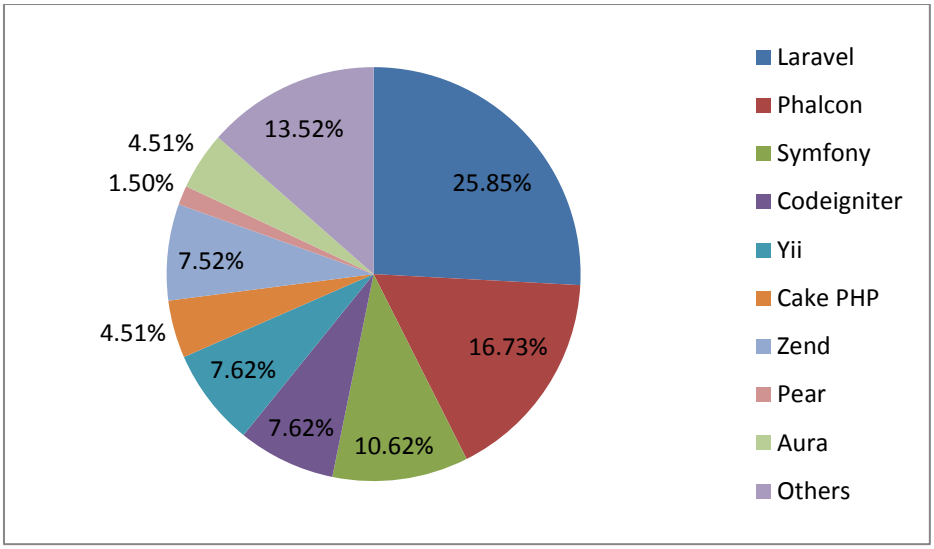

Figure 8 : Compare among all MVC-PHP framework based on current trends.

From Fig 8, it is shown clearly that, laravel took almost 26% place in world web development in 2013 by MVC pattern framework. Based on this analysis, it is understandable that, laravel is going to be one of the most famous MVC pattern framework in PHP world with huge flexibility of deployment as well as manage.

4. Conclusion:

An empirical study on major MVC pattern for PHP framework has been evaluated in this paper.

. The results obtained from evaluating the four frameworks: CodeIgniter (CI)Symfony CakePHP and Laravelusing criteria such as request per second, system load average, memory usage, number of function calls and number of files required as well as available facilities in each framework shows that theLaravel outperforms other MVC framework. A request per second of as high as 3000 was recorded forLaravelcompare to others like CakePHP with as low as 750 request per second. The results obtained for all other parameters such has storage, function calls, number of files, response time e.t.c. indicates thatlaravelhas huge flexibility of development of web application, it has some more facilities for programmers that makes it acceptable to all web programmer in terms of different criteria such as intuitive, compiled fast, cross platform, open source, flexibility. It enable easily migration, enriched library, template system, eloquent ORM and wide range of community support that helps to develop application smoothly. All the criteria and facilities of laravel prove that, it is of opinion that laravel would be the best choice to deploy next generation PHP based web application

References:

[1] Shin Nakajima, KeijiHokamura&NaoyasuUbayashi, Aspect-Oriented Development of PHP-based web applications, 34th Annual IEEE Computer Software and Applications Conference Workshops, 2010

[2] S. Bergmann and G. Kniesel, GAP: Generic Aspects for PHP, In Proc. EWAS'06, 2006

[3] F. Ricca and P. Tonella, Analysis and Testing of Web Applications. In Proc. 23rd ICSE, pages 25-34, May 2001.

[4] StigSaetherBakken, Alexander Aublach, EgnnSchmid et al, PHP Manual (The PHP Documentation Group)

[5] Veglis A, Leclercq M, Quema V, PHP and SQL made simple Distributed Systems Online, Aug 2005.

[6] Wei Cui, Lin Huang, LiJing Liang and Jing Li, The Research of PHP Development Framework Based on MVC Pattern, 4th International Conference on Computer Sciences and Convergence Information Technology, 2009.

[7] http://www.sitepoint.com/the-mvc-pattern-and-php-1/ (access date: 15 January 2014)

[8] http://www.tonymarston.net/php-mysql/model-view-controller.html(access date: 13 December 2013)

[9] http://www.sitepoint.com/application-development-cakephp/(access date: 17 January 2014)

[10] http://www.developed.be/2013/07/16/php-frameworks-which-to-choose/(access date: 16 January 2014)

[11] http://webcoderpro.com/blog/top-6-most-popular-php-frameworks-of-2013/(access date: 21 January 2014)

[12] http://www.sitepoint.com/best-php-frameworks-2014/(access date: 20 January 2014)

[13] http://www.catswhocode.com/blog/top-10-php-frameworks-for-2014(access date: 23 January 2014)

[14] http://brianretterer.com/why-laravel-is-the-best-php-framework/(access date: 15 January 2014)

[15] http://www.webdesignermag.co.uk/features/laravel-a-modern-php-framework/(access date: 17 January 2014)

[16] http://matrix.include-once.org/framework/simplese(access date: 19 January 2014)

[17] Ivan Enderlin, Alain Giorgetti and Fabrice Bouquet, A Constraint Solver for PHP Array, 6th International Conference on Software Testing, Verification and Validation Workshops, 2013

[18] Ettore Merlo, Dominic Letarte and GiulianoAntoniol, Automated Protection of PHP Applications against SQL-injection attacks, 11th European Conference on Software Maintenance and Reengineering, 2007

[19] http://en.wikipedia.org/wiki/MIT_License (access date: 28 February 2014)

[20] http://www.php.net/manual/en/intro.xhprof.php (access date: 10 March 2014)

[21] http://www.ruilog.com/blog/view/b6f0e42cf705.html (access date: 8 March 2014)

[22] http://en.wikipedia.org/wiki/CodeIgniter (access date: 10 March 2014)

[23] http://en.wikipedia.org/wiki/Symfony (access date: 11 March 2014)

[24] http://en.wikipedia.org/wiki/Laravel_(framework) (access date: 9 March 2014)