SELECTING A PHP FRAMEWORK FOR A WEB APPLICATION PROJECT — THE METHOD AND CASE STUDY

Conference Paper · March 2015

DOI: 10.13140/RG.2.1.1321.2567

2 authors:

Adrian Zurkiewicz
Wroclaw University of Science and Technology
1 PUBLICATION 3 CITATIONS
SEE PROFILE

Marek Miłosz
Lublin University of Technology
203 PUBLICATIONS 329 CITATIONS
SEE PROFILE

SELECTING A PHP FRAMEWORK FOR A WEB APPLICATION PROJECT — THE METHOD AND CASE STUDY

A. Zurkiewicz, M. Milosz

Lublin University of Technology (POLAND)

Abstract

Considering any PHP project in terms of its purpose and requirements, the choice of the most appropriate framework significantly affects the cost of a project and time necessary for designing and creating it, as well the likelihood of its successful completion. However, it is a very difficult choice to make, because it requires conducting a multi-criteria comparison among numerous available frameworks. This article presents a multi-criteria method of selecting a PHP framework for a PHP web application designed for university environment and compares selected frameworks with the goal of designing and carrying out the project.

There are many multi-criteria comparison methods, and choosing the appropriate one affects the possibility of conducting comprehensive research and the clarity of results. Another crucial comparison factor consists in selecting adequate comparison criteria and adjusting their respective weights. In the article, the choice of a comparison method as well as its criteria and weights are discussed and justified. Moreover, the values of measurable criteria are determined in order to compare in the clearest possible way the features of all the analyzed frameworks with regard to the specified project, and finally, the results of the comparison are presented. The analysis includes the following frameworks: CakePHP, CodeIgniter, Laravel, Symfony, Yii, and Zend Framework.

The article describes an actual web application project named Expansible Communication Platform for e-University (XCPU), which was designed to improve communication between the university authorities, teachers and students in order to gain information for the decision-making process. The XCPU project is described with its requirements and characteristic conditions. Together with the results of the case study of selecting a proper PHP framework in the context of the described web application project, the article also presents the detailed procedure of framework selection, the algorithm and the calculations with their results.

Keywords: e-university communication, web application technology, frameworks, selection method

1 INTRODUCTION

In terms of how applications are developed, the web application industry can be divided into two main categories: the applications created on the basis of ready-made components or modules and the ones that are created from scratch and customized for the needs of each customer Using ready-made components allows creators to develop applications quickly and reduces the initial costs of a project, which is crucial as far as low-budget applications are concerned. However, there are three major flaws of such solution: 1) even though the creators of ready-made solutions pay great attention to ensure that the final product can be customized and modified, there are many obstacles that prevent adjusting the applications to the customer's needs; 2) the cost of subsequent modifications grows exponentially due to the incompatibility of individually versioned elements; 3) openness of the source code limits the security of web applications. An alternative approach to web application development consists in creating an application from scratch for the individual needs of a customer. Such approach might help with overcoming the aforementioned problems, but the costs and time necessary for the application development will increase. There is also a mid-way solution: creating a web application based on the already existing frameworks, which provide an organizational and functional base for the whole project, regardless of its end-use. Using a framework allows the developer to significantly reduce the costs and time necessary for completing the project without limiting its functionality.

Although frameworks can be seen as a universal base for a project, selecting the framework affects the cost and time needed for completing the project. Also, due to efficiency issues and the character of the industry, the framework selection affects the costs of application maintenance, particularly the demand for the resources of a production server. Consequently, it becomes crucial to be able to quickly and reliably compare the available frameworks in terms of their usefulness for developing a new web application.

2 COMPARATIVE ANALYSIS METHODS

The purpose of a comparative analysis is to evaluate and classify complex elements on the basis of a specific feature or a collection of features. Such analysis is useful for identifying the differences between the analyzed objects or ranking them and is very commonly used in the decision-making process. When various features affecting the evaluation of the object are analyzed, one often receives a series of sub-scores containing data in various scales and of various impact on the evaluation. There is a distinction between three types of diagnostic variables [1]:

- **Stimulants** — the variables whose higher value means that the target object has better parameters in terms of the examined criterion
- **Destimulants** — the variables whose lower value indicates more desirable properties of the target object.
- **Nominants** — the variables whose specific value indicates a desirable property of the target object.

Normalization formulas are used for normalizing diagnostic variables whereas non-negative weights can be applied to measure the impact of the given criterion on the analysis [1].

Having a set of normalized-value features, one can rank the objects on the basis of a synthetic feature by means of the following methods:

- **Non-reference method** — consists in creating a new synthetic feature by summing up the weighted, normalized value of sub-features. The highest value of a synthetic feature indicates an object best fulfilling the criteria of analysis [2].
- **Reference method** — consists in determining the optimal values of features and creating a reference object on their basis. Then, the metric space distance is determined between each analyzed object and the reference object. The objects located closest to the reference object fulfill the criteria of analysis in the best way. The most commonly used metric is Euclidean metric [1],[3].

3 THE APPLIED METHOD OF COMPARATIVE ANALYSIS

With regard to research into using frameworks in XCPU (Expansible Communication Platform for e-University) project, all the criteria of comparative analysis are either stimulants or destimulants. Consequently, there is no need for creating a reference set of criteria and, as a result, the non-reference method of ranking on the basis of a synthetic feature allows one to carry out a quick and relatively unambiguous comparative analysis. Formulas 1 and 2 enable one to normalize feature values whereas formula 3 enables one to calculate the value of a joint synthetic criterion [4],[2].

$$Z^S = \frac{X^S - X_{min}}{X_{max} - X_{min}} \tag{1}$$

$$Z^S = 1 - \frac{X^D - X_{min}}{X_{max} - X_{min}} \tag{2}$$

$$K = \sum_{i=1}^{n} w_i \cdot Z_i^S \tag{3}$$

Where:

Z^S = Normalized value (of a stimulant).
X^D = Observed value (of a destimulant).
X^S = Observed value (of a stimulant).
X_{min} = Minimum observed value.
X_{max} = Maximum observed value.
K = Joint synthetic criterion.
w_i = Weight of a sub-criterion of i-feature.
$Z^S{}_i$ = Normalized value of a sub-criterion of i-feature (of a stimulant).

The procedure of ranking on the basis of the non-reference synthetic method consists of the following steps:

1) Determination of the criteria on the basis of which the comparative analysis will be conducted.

2) Determination of the weights of particular criteria, for example on the basis of a substantive evaluation performed by an expert.

3) Research into determining the values of features within the scope of the criteria for the analyzed objects.

4) Normalization of the value of the observed object features by means of formulas 1 and 2.

5) Establishing a joint synthetic criterion by means of formula 3.

6) Arrangement of the analyzed objects on the basis of the joint synthetic criterion.

4 CRITERIA FOR USEFULNESS EVALUATION

There are two groups of factors that affect how useful a framework is for creating web applications: the factors affecting the source code or its performance and the factors that affect the work of programmers and software developers, but don't have a direct impact on the source code and how an application works. The first group comprises the efficiency of a framework and conciseness of source code. Non-direct factors include quality of documentation and its availability, availability of technical support and integration of tools that support the process of creating web applications.

4.1 Framework documentation

Documentation is a crucial factor, which affects the work of programmers and software developers. Documentation enables creators to quickly evaluate a framework's potential and determine the conditions that have to be fulfilled so that all its functionalities can be used and to ensure that the application works correctly and securely. Well-written documentation should consist of two parts. The first is an external part comprising an indexed collection of all the documents, which is usually available directly on the website of a framework's creator. The second part comprises properly formatted documentation of all the individual elements on the framework source-code level, which enables the developer to make use of the prompts from advanced, modern programming environments such as Eclipse or NetBeans IDE.

4.2 Tools supporting web application development

A substantial part of the source code of modern web applications consists of repeatable and standard elements. For this reason, many modern frameworks are distributed together with tools that support the development of such source code elements. Such tools considerably speed up the process of developing the source code of an application, reduce the programmer's workload and decrease the risk of him or her making a programming mistake.

The tools that support web application development, in particular the ones based on the MVC pattern, can support the development of the following elements of source code:

- **Models** in the MVC pattern represent a problem subject to business logic. In the context of web applications, such models are very often responsible for table mapping, making it possible to acquire and manipulate the data from tables.
- **CRUD** (create, read, update and delete records) tools enable one to perform basic operations on the records in a database.
- **Controller** classes are responsible for the functionality of a controller in the MVC pattern.
- **Views** are an element of the MVC pattern.
- **Forms** are usually listed as separate, single classes or groups of classes responsible for rendering and validation logic of entry data.
- **Modules** are not present in every framework, but they are very useful and can be effective in separating functionality sets and ensuring portability of the source code between projects.

4.3 Supported technologies and programming techniques

Even though frameworks support many technologies and programming techniques, the extent of that support varies, which might be a key factor regarding the usefulness of a specific solution and choice of a framework.

In the context of the article, programming techniques are understood as identifiable and named solutions to common programming problems [5]. In the analysis the following programming techniques are considered:

- **MVC** (Model-View-Controller)
- **ORM** — Object-Relational Mapping - the technique that consists in mapping of the database objects and the relationship between them and programming language classes.
- **DB Objects** support the access to a database by means of specialized classes.
- **Templates** support the techniques that enable the creator to manage the templates such as: *two-Step view* or *composite view* [6],[7].
- **Caching**, the technique of storing the intermediate results, can be provided by various external technologies, but the modern frameworks provide classes operating on the basis of the facade pattern so that they can provide the same work interface independent of the library in use.
- **AJAX** — frameworks providing classes that increase performance and simplify source code.
- **Modules** enable the programmer to divide source code into functionality groups, which improves management and the portability of the source code between projects.
- **EDP** (Event Driven Programming) is a technique of source code organization on the basis of the events that occur while serving a request.
- **Namespaces** — provides better source code management.

The technologies that have a significant impact on the usefulness of a framework in the context of a web application project: 1) support for various database management systems; 2) support for various buffering mechanisms (caching); 3) support for integration: SOAP and simultaneous support of various data sources.

4.4 Conciseness of source code

Even though conciseness of source code understood as labor intensiveness of programming is quite hard to evaluate, it is crucial in terms of the economic context of a web application project. Three factors are distinguished as a basis for labor intensiveness evaluation: the number of lines in source code, the average length of a source code line and the number of source code files. Such approach provides a simplified but quite objective view of source code conciseness.

4.5 Framework efficiency

Framework efficiency is significant in terms of ensuring the optimal performance of an application as far as limited production server capabilities are concerned. There are two main performance parameters which describe the efficiency of a web application from the end-user point of view, and in the context of the article, they describe the efficiency of the framework itself: 1) maximum time required for serving the request; 2) the number of requests than can be served in a specific unit of time.

5 CASE STUDY

The aim of the comparative analysis carried out as a part of this article is to choose the most appropriate PHP framework to create a web application named *Expansible Communication Platform for e-University* (XCPU). The application facilitates communication between university authorities, teachers and students in order for them to acquire the information used in decision-making process. The description of the functional and non-functional requirements of XCPU includes mainly the elements that affect the choice of comparison criteria and their weights and ultimately the course of analysis.

5.1 Functional requirements

The platform has to provide the functionalities related to user account administrations and the rights to access specific modules from the level of individual users and user groups. Additionally, the platform has to provide the mechanism of invitations that give access to the forms that collect information from the users that are not logged in or do not have an account. The platform also has to provide mechanisms for e-mail communication with users and invited respondents. It should also provide tools for creating PDF reports together with the mechanism of graphical data representation through defined diagrams, tables and graphs. Other functionalities should be carried out by independent application modules:

- **Quality assessment module** — allows creating surveys directed to the groups of university employees and students, thus enabling them to assess various aspects of the way the university works.
- **Opinion survey module** — allows for creating thematic opinion surveys directed to the groups of university employees and students.
- **Course instructors' evaluation module** — allows students to evaluate their course instructors, considering multiple, programmable evaluation criteria.
- **Curriculum evaluation module** — enables students and course instructors to evaluate the curriculum of each course, considering multiple, programmable criteria.
- **Students assessment module** — allows for assessing a student's performance in each course
- **Ideas module** — allows the platform users to gather ideas regarding the way the university functions. The users can also address proper administrative units and track the life-cycle of an idea.
- **Project monitoring module** — allows for the administration of university projects together with the elements of progress monitoring, description of each stage and information on who is responsible for the project and involved in it.

5.2 Non-functional requirements

The platform has to allow independent operation of each module and possibility of adding more modules while the application is used. Modules should be independent to the greatest possible extent so that the application can operate after the basic modules are loaded, which should ensure progressive structure of the application. It has to be ensured that the application can be integrated with already existing elements of a university's IT system, which makes it necessary for a framework to support integration technologies.

6 1 COMPARATIVE ANALYSIS OF FRAMEWORKS

For the purposes of this research, the following factors have been taken into account when choosing frameworks to analyze: 1) a framework's popularity represented on figure 1; 2) source-code openness of the software.

The following frameworks have been chosen: CakePHP (version 2.5.3), CodeIgniter (version 2.2.0), Laravel (version 4.2), Symfony (version 2.5.3), Yii (version 1.1.15) i Zend Framework (version 2.3.2).

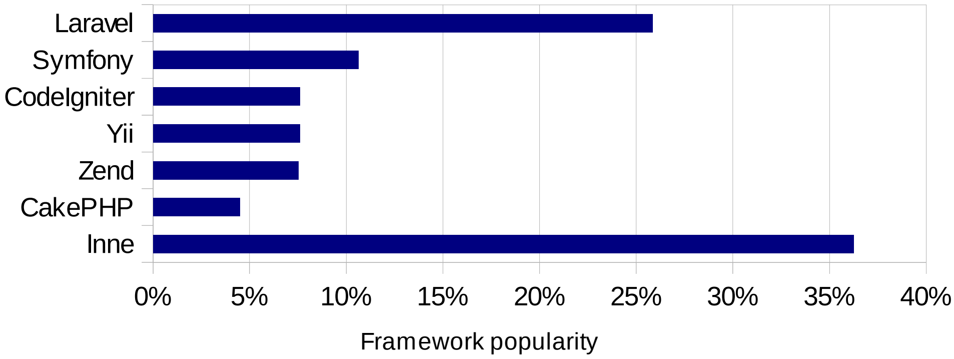

Figure 1. The most popular frameworks in 2013
Source: Own elaboration based on [8]

6.1 Parametrization of the comparative analysis

In the context the XCPU web application project, the weights used in the comparative analysis are intended to match the parameters of analysis to the described choice of framework in the best possible way. As a consequence, it becomes possible to determine the best framework for developing the described web application. Table 1 shows the weights used in the comparative analysis carried out in this work.

Table 1. Weights for the criteria

Criterion	Description	Weight
Documentation and technical support	They are the key factors for ensuring source code reliability.	0.10
Tools supporting web application development	They can reduce programmers' workload, but their usefulness is relatively limited due to overly general nature of generated source code.	0.05
Programming techniques	The can significantly reduce the amount of source code, which makes the application more reliable.	0.15
Database technologies	Their significance consists in allowing applications to integrate with other elements of a university IT system. However, each framework supports the most commonly used database technologies.	0.10
Caching (Buffering)	Caching is significant because it ensures the optimum performance of the application. However, all the frameworks in the comparison support at least the most commonly used caching mechanisms.	0.05
Integration	Because it is necessary for a framework to be compatible with other elements of a university's IT system, integration is a very important factor in developing the XCPU platform.	0.30
Conciseness of source code	Its significance consists in reducing the possibility of making mistakes, so it improves the reliability of application. Moreover, reducing labor-intensiveness of programming significantly affects the cost of developing an application, and in consequence, it reduces the final cost of the XCPU project.	0.10
Framework efficiency	Efficiency can be of critical importance due to the strictly determined hourly work schedule of universities, which might result in very uneven server workload and occurrence of periods when its resources are used very intensively.	0.15

6.2 Framework comparison

Table 2 shows the rules of scoring for framework documentation. With regard to the contextual source documentation conforming to the PHPDoc format, documentation is considered to be complete when it describes all the occurrences of the element and incomplete when it includes only some occurrences. The table also shows the way dynamic programming techniques affect access to the contextual documentation of the source code when working with controller and model classes. The evaluation of contextual documentation has been normalized by means of formula 1.

Table 2. Criteria for documentation scoring

Category	Feature	Value	Points
General documentation (1 pt. maximally)	English		0.50 pt
	Number of additional languages	1-4	0.25 pt
		5-10	0.40 pt
		>10	0.50 pt
Contextual documentation (3 pt. maximally)	Classes	Complete	1.00 pt
		Incomplete	0.50 pt
	Class methods	Complete	1.00 pt
		Incomplete	0.50 pt
	Class properties	Complete	1.00 pt
		Incomplete	0.50 pt
	Dynamic coding (controller)	Limited impact on work	-0.25 pt
		High impact on work	-0.50 pt
	Dynamic coding (model)	Limited impact on work	-0.25 pt
		High impact on work	-0.50 pt

Table 3 shows the result of documentation analysis for each framework. The "Result" column shows the normalized sum of points in each category.

Table 3. Framework documentation

	Translation of documentation (number of languages)	Documentation (points)	Code documentation (points)	Result
CakePHP	6	0.90	0.71	**1.00**
CodeIgniter	1	0.50	0.57	**0.51**
Laravel	1	0.50	0.00	**0.00**
Symfony	1	0.50	1.00	**0.90**
Yii	15	1.00	0.14	**0.58**
Zend Framework	1	0.50	0.71	**0.64**

6.3 Analysis of tools supporting web application development

The analysis of tools supporting web application development has been conducted on the basis of the documentation provided by the framework supplier and on the basis of the tests carried out during the analysis preparation. The evaluation is binary: 0 - not supported; 1 – supported.

Table 4 presents the frameworks in terms of their compatibility with tools supporting web application development. The "Result" column shows the normalized sum of points.

Table 4. Tools supporting web application development

	CakePHP	CodeIgniter	Laravel	Symfony	Yii	Zend Framework
Models	1	0	0	1	1	0
CRUD tools	0	0	0	1	1	0
Controller	1	0	1	1	1	0
Views	1	0	0	0	0	0
Forms	0	0	0	1	1	0
Modules	1	0	0	1	1	0
Result	**0.80**	**0.00**	**0.20**	**1.00**	**1.00**	**0.00**

6.4 Analysis of supported programming techniques

The analysis of supported techniques was conducted on the basis of the documentation provided by the framework supplier and on the basis of the tests carried out during the analysis preparation.

Table 5 presents the frameworks in terms of the techniques they support. A framework is awarded only half a point if it only partially supports a given technology. The "Result" column shows the normalized sum of points.

Table 5. Supported techniques and technologies

	CakePHP	**CodeIgniter**	**Laravel**	**Symfony**	**Yii**	**Zend Framework**
MVC	1	1	1	1	1	1
ORM	0	0	1	1	1	0
DB Objects	1	1	1	1	1	1
Templates	1	1	1	1	1	1
Caching	1	1	1	1	1	1
Ajax	1	1	1	1	1	1
Modules	1	0	1	1	1	1
EDP	0	0	1	0	1	1
Namespaces	0	0	0.5	1	0	1
Result	**0.29**	**0.00**	**1.00**	**0.86**	**0.86**	**0.86**

6.5 Analysis of supported external elements

The analysis of supported databases has been conducted on the basis of framework documentation. The highest possible score was given for the support of the PDO technology, as it provides access to a wide range of databases. Although some frameworks perform the support by means of PDO drivers, they need special external layers that reduce the number of available databases to the ones included by the developer during the framework's development. In such cases, points are awarded only for the support for individual database systems, but not for PDO.

Table 6 shows the database systems supported by each framework. The "Result" column shows the normalized sum of points.

Table 6. Supported database systems

	CakePHP	**CodeIgniter**	**Laravel**	**Symfony**	**Yii**	**Zend Framework**
MySQL	0.25	0.25	0.00	0.25	0.00	0.25
PostgresSQL	0.25	0.25	0.00	0.25	0.00	0.00
Sqlite	0.25	0.25	0.00	0.25	0.00	0.00
MS Sql Server	0.25	0.25	0.00	0.25	0.00	0.25
Oracle	0.00	0.25	0.00	0.25	0.00	0.00
PDO	0.00	1.00	1.00	0.00	1.00	1.00
ODBC	0.00	0.50	0.00	0.00	0.00	0.00
Result	**0.00**	**1.00**	**0.00**	**0.14**	**0.00**	**0.29**

Table 7 shows buffering technologies supported by each framework. The "Result" column shows the normalized sum of points.

Table 7. Supported buffering technologies

	CakePHP	**CodeIgniter**	**Laravel**	**Symfony**	**Yii**	**Zend Framework**
APC	1.00	1.00	0.00	1.00	1.00	1.00
Memcached	1.00	1.00	1.00	0.50	0.00	1.00
Memcache	1.00	0.00	0.00	0.50	1.00	1.00
Redis	1.00	0.00	1.00	0.50	1.00	1.00
Wincache	1.00	0.00	0.00	0.50	1.00	1.00
Xcache	1.00	0.00	0.00	0.50	1.00	1.00
Files	1.00	1.00	0.00	1.00	1.00	1.00
Result	**1.00**	**0.20**	**0.00**	**0.50**	**0.80**	**1.00**

6.6 Comparison of integration support

The analysis of supported integration technologies has been conducted on the basis of documentation and source code analysis. The most emphasis has been put on the tools and libraries integrated with the base version of a framework

Table 8 shows integration technologies supported by each framework. The "Result" column shows the normalized sum of points.

Table 8. Supported integration technologies

	CakePHP	CodeIgniter	Laravel	Symfony	Yii	Zend Framework
SOAP	0	0	0	0	1	1
Access to many databases	1	1	1	1	1	1
Result	**0.0**	**0.0**	**0.0**	**0.0**	**1.0**	**1.0**

6.7 Analysis of source code conciseness and web application efficiency

The analysis of source code conciseness and framework efficiency has been conducted on the basis of a specially prepared application. The application has been coded in six versions, each version created by means of a different framework. The application was supposed to perform two functionalities: 1) displaying a list of students based on the table from the database; 2) adding students to the table on the basis of the form including the validation mechanism.

Table 9 shows the result of code conciseness analysis. The "Points" column shows the normalized sub-score, whereas the "Result" column shows the normalized joint result for the "Code conciseness" category.

Table 9. Code conciseness

	Lines of code	Points	Average length of line code	Points	Number of files	Points	Result
CakePHP	60	1.00	28	0.40	6	1.00	**1.00**
CodeIgniter	83	0.90	30	0.20	6	1.00	**0.79**
Laravel	100	0.83	25	0.70	7	0.83	**0.97**
Symfony	123	0.73	32	0.00	9	0.50	**0.16**
Yii	119	0.75	23	0.90	9	0.50	**0.82**
Zend Framework	292	0.00	22	1.00	12	0.00	**0.00**

Table 10 shows the result of efficiency analysis for each analysed framework. The "Request serving time" column shows the maximum time necessary for serving the client, considering 99% of the most quickly served requests. The "Requests per second" column shows the average number of requests served by the server during one second. The "Points" column shows the normalized sub-score, whereas the "Score" column shows the normalized joint result for the "Conciseness of the code" category. The research was performed on a sample of 10,000 requests for each framework.

Table 10. Framework efficiency

	Request serving time (ms)	Points	Requests per second	Points	Result
CakePHP	462	0.49	130	0.20	**0.35**
CodeIgniter	50	1.00	480	1.00	**1.00**
Laravel	389	0.58	102	0.14	**0.36**
Symfony	859	0.00	42	0.00	**0.00**
Yii	97	0.94	229	0.43	**0.68**
Zend Framework	187	0.83	108	0.15	**0.49**

6.8 Comparative analysis result

Figure 2 shows the final result of the comparative analysis of the chosen frameworks on the basis of the comparative model proposed in this article. The "Result" axis shows the sum of all the results achieved by each analyzed framework in terms of the criteria of analysis and considering their weights. The result is calculated by means of formula 3.

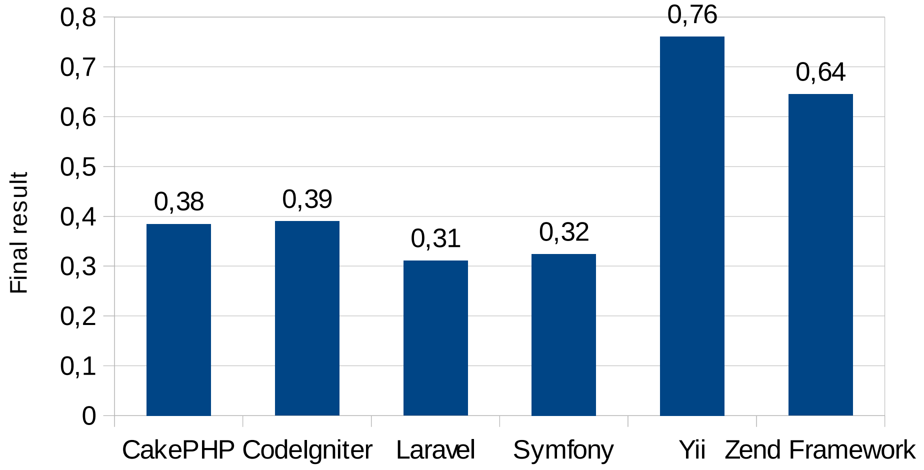

Figure 2. Result of the comparative analysis of the chosen frameworks

Source: Own elaboration

7 CONCLUSIONS

The analysis shows that Yii and Zend are the most appropriate frameworks for creating the described web application. However, the comparative analysis of the chosen criteria shows that other frameworks can be more very useful for some specific purposes. For example, CodeIgniter proved to be very useful for creating applications that work under a heavy burden. This example shows how crucial it is to choose appropriate criteria and weights for the comparative analysis in order to choose the framework for the purposes of a specific project.

REFERENCES

[1] Langley, R. (1971). Practical Statistics Simply Explained, Dover Publications, p. 399

[2] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review, ACMComputing Surveys, vol. 31, iss. 3, pp. 264-323.

[3] Cluster Analysis (2014). Available online at: http://www.statsoft.com/Textbook/Cluster-Analysis

[4] Larsen, R., and Marx M. (2011). An Introduction to Mathematical Statistics andIts Applications, 5th Edition, Pearson, p. 768.

[5] PHP Frameworks (2014). Available online at: http://www.phpframeworks.com/

[6] Two Step View (2014). Available online at: http://www.martinfowler.com/eaaCatalog/twoStepView.html

[7] Composite View (2014). Available online at: http://www.corej2eepatterns.com/Patterns2ndEd/CompositeView.htm

[8] Best PHP Frameworks for 2014 (2014). Available online at: http://www.sitepoint.com/best-php-frameworks-2014