



ADOPTING WEB FRAMEWORK IN WEB-APPLICATION DEVELOPMENT

Case: Kehittäjän tieto- ja menetelmäpankki

Lahti University of Applied Sciences

LAHTI UNIVERSITY OF APPLIED
SCIENCES

Degree Programme in Business

Information Technology

Bachelor's Thesis

Spring 2012

Nguyen Minh Thanh

Lahti University of Applied Sciences
Degree Programme in Business Information Technology

NGUYEN, MINH THANH: Adopting web framework in web
application development
Case: Kehittäjän tieto - ja
menetelmäpankki

Bachelor's Thesis of Degree Programme in Business Information Technology, 40
pages, 3 pages of appendices

Spring 2012

ABSTRACT

Since the web found its true form, it is no longer static web pages without any user's interaction. During the last few years, web applications have matured to the point that they can compete with full-fledged desktop applications. The support technologies have been growing significantly with the likes of Google web toolkit, web application frameworks, and CMS.

This thesis is based on the real application of Tykes, which has been coded by hand in PHP for a few years. The approach is set to improved Tykes's features and performance. As the result, it shows the benefits of implementing web frameworks in real life works.

The action research method is used to answer the research questions in this study, in inductive approach. The researcher's experience is compared to other relevant published sources. Example source codes are extracted from Tykes.

The research results show an expected range. It confirms the positive effects of implementing web framework in web application development. Features which have been tested in Tykes are presented and compared to their originals. Some example source codes are extracted from the application to prove the results.

To conclude, the reasons and recommendations are put forward. Although the frameworks and their communities have grown up dramatically and are near maturing, there are still lacking of features and incompatibility problems. Therefore, further study is recommended on those.

Keywords: web application, web framework, PHP

CONTENTS

1	INTRODUCTION	1
1.1	Research background	1
1.2	Research questions and research objectives	2
1.3	Research approach	3
1.4	Research Methodology	3
1.5	Scope and limitations	4
1.6	Thesis structure	5
2	INTRODUCING WEB FRAMEWORK	7
2.1	Web Framework definition and characteristics	7
2.2	Web Framework in comparison	9
2.2.1	Wed Framework against Library	9
2.2.2	Framework against CMS and WYSIWYG	10
2.3	Design pattern of Web Framework	11
3	WEB APPLICATION LANGUAGES	16
3.1	PHP among other languages	18
3.2	Introduce some popular PHP framework	19
3.2.1	Zend Framewok	20
3.2.2	CakePHP	21
3.2.3	Yii	22
3.3	Comparison between the frameworks	23
4	FEATURES COMPARISON	25
4.1	Model View Controller	25
4.2	Learning curve	27
4.3	Database	28
4.3.1	Connection	29
4.3.2	Mapping and Methods	30
4.4	Security	31
4.5	Internationalization and localization	33
4.6	User management	34
5	CONCLUSIONS	38
	REFERENCE	41
	APPENDICES	44

LIST OF FIGURES

FIGURE 1: Example code in PHP	8
FIGURE 2: Technical knowledge needed (Bergeret, 2010).....	11
FIGURE 3: MVC model in SmallTalk80 (Krasner & Pope, 1988)	12
FIGURE 4: Define database link in Zend Framework	12
FIGURE 5: Controller_The application logic	13
FIGURE 6: View_The application presenter	13
FIGURE 7: MySQL database table	13
FIGURE 8: Model's methods used in application controller.....	14
FIGURE 9: MySQL table's structure.....	15
FIGURE 10: Diagram for decision making (Synodinos, 2007)	17
FIGURE 11: Job Trends for Zend, CakePHP and Yii (Job trends, 2012).....	21
FIGURE 12: PHP frameworks performance in comparison (Framework performance comparison)	23
FIGURE 13: Result for "Which framework is the most promising?"	24
FIGURE 14: Typical Flow in PHP Programs (Golding, 2008).....	26
FIGURE 15: Data Flow in CodeIgniter Programs (CodeIgniter User Guide, 2012)	26
FIGURE 16: Database connection in PHP without framework	30
FIGURE 17: Data input in PHP without framework.....	30
FIGURE 18: CSRF hash implementation in Zend	31
FIGURE 19: Form controller in Zend	32
FIGURE 20: CSRF has implementation without framework.....	32
FIGURE 21: Form validation without framework	32
FIGURE 22: Internationalization and localization are orthogonal to the application layers (Sachdev, 2007).....	33

LIST OF TABLES

TABLE 1: Programming Language position in 1996, 2006 and 2011 (Tiobe programming community index, 2012)	18
TABLE 2: Features comparison	36

GLOSSARY

WWW	World Wide Web
HTML	Hyper Text Markup Language
PHP	PHP: Hypertext Preprocessor
CMS	Content Management System
WYSIWYG	What You See Is What You Get
MVC	Model-View-Controller
SQL	Structured Query Language
LAMP	Linux-Apache-MySQL-PHP
RMDS	Relational Management Database System
CSRF	Cross-Site Request Forgery
I18N	Internationalization
L10N	Localization

1 INTRODUCTION

Originally, the World Wide Web was imagined as a means for sharing information where documents are linked together in an inter-network (Berners-Lee & Cailliau, 1990). These documents were basically static. As growing, forms were introduced, helping users to interact with servers. Soon after, the first Web application was created with the born of server-side scripting language making dynamic generation of HTML documents possible. Until now, the web technology has reached to a new standard where web applications have full abilities of desktop applications. A group of dynamically-generated web pages is the most common way to represent web applications.

The online market is emerging required developers to produce better featured application within less amount of time. Web application frameworks are created with this purpose in mind. As companies have applied and succeeded in saving time and boosting applications' features, frameworks have proven its ability to cope with the market demands.

1.1 Research background

In October 2011, a web server survey made by Netcraft, an England based Internet Services Company, received responses from 504,082,040 sites. According to the survey, there are around 506 million websites across all domains, and has increased significantly from 255 million websites as of December 2010 (October 2011 Web Server Survey, 2011). Another study made by Pingdom, a Sweden based website monitoring company, in December 2010 shows many other interesting stats, email for example. People are getting more and more familiar with this electronic mail which is faster, and costs much less than the original written letter. To illustrate it, 107 trillion is the number of emails sent on the Internet in 2010, so we have the average number of 294 billion emails per day during 2010 (Internet 2010 in numbers, 2011).

Internet technology is moving forward surprisingly fast. It offers great support to upgrade living conditions of human beings. People can do almost everything with their computer with support of the Internet, with the increasing number of data

and services available over the Internet. It includes most everyday activities. In the entertainment industry, DVD is one of the most used instruments to distribute media. However, according to an analysis from Goldman Sachs', Netflix, a US cable company having most subscribers, would end its DVD distribution by 2020 (Social media war for video, 2011). Besides, with the rise of online services such as Amazon, even fresh products can be delivered to home everyday when needed.

In this theme, creating a web application is a vital task for every company.

Everyone may know web applications have something in common. Most of them have users who can register, gain access and interact with the application.

Interactions are mostly carried out through preset and secure forms, connected to various databases. Databases are then searched to process data and presented back to the user. Grouping these patterns into some kinds of abstraction and transport them further into other applications could push up the process much faster. Many different ways have been applied, such as CMS, Libraries, WYSIWYG and Frameworks. This is why there are so many brilliant choices to choose for developers to start working on a new web application to make it better and faster. Above all, Web Framework is proving to be one of the most emerging platforms to support web application development.

1.2 Research questions and research objectives

Given the above information, the research questions of this study are:

- What are the reasons for new developers choose to work with framework?
- In what case should developers apply framework?

This thesis introduces Zend, one of the most anticipated web frameworks, to illustrate frameworks' usage. Since the release of Ruby on Rail in 2004, full-stack framework has proved its ability to aid web application development. Many different framework projects for different programming language have been launched. Until the time of this study, stable versions are available for many frameworks and new features are being tested via their beta versions. For that reason, more and more developers, including free-lancer developers and even

larger groups and companies, choose framework as their start-up tool. The goal of this thesis is to provide the insight of web framework, how it could be used to boost the development process. This thesis will be of interest to many new or soon-to-be web application developers to decide whether framework is the right tool to further their careers.

1.3 Research approach

According to Prof. Burney, in logic, there are two broad methods of reasoning as referred to the deductive and inductive approaches. Sometimes, the two methods are combined to have the third kind. Deductive approach moves from the more general to the more specific which usually begin with the theory then narrow down to the specific problem that we can test and conclude the research with a confirmation. Inductive approach, on the other hand, works the other way. It begins with some specific observations to detect the patterns and ends up with broader theory. (Burney, 2008)

In this study, inductive is chosen as the approach because of the characteristics of the research problems as well as the research methodology, which will be mentioned in the following part. Following inductive approach, the researcher first studies the development process of Tykes by applying Zend Framework. By observing the process, different between methods, which are manually coding and coding in framework, will be analyzed to give a broader generalization of web framework and answer the above research questions.

1.4 Research Methodology

The aim of this thesis is to study and understand in-depth the benefit of implementing framework in web application development. It requires a thorough acknowledgment of the whole process understanding all influence factors. To obtain the result, this thesis uses action research as its main research methodology.

“Action research...aims to contribute both to the practical concerns of people in an immediate problematic situation and to further the goals of social science simultaneously. Thus, there is a dual commitment in action research to study a

system and concurrently to collaborate with members of the system in changing it in what is together regarded as a desirable direction. Accomplishing this twin goal requires the active collaboration of researcher and client, and thus it stresses the importance of co-learning as a primary aspect of the research process.” (Gilmore, Krantz, & Ramirez, 1986)

Originally founded by Kurt Lewin, action research is known simply as “learning by doing”. Researchers try to approach the problem and attempt to resolve it. Their effort is collected and analyzed during the process.

To cope with the growing pace of Internet technology and user’s needs, web based applications need to become more and more flexible and can do many more things. The application would get out-dated easily without frequent update and ability to scale up to meet up with demands.

Together with action research, qualitative research tools will be used to gain more views from developers and companies who currently employ or are in the process of employing frameworks in their web application.

Due to the similarity of the two research methods, qualitative and action research, and characteristics of the action research, which are more about problem-solving than simple collecting and analyzing of data, Rory O’Brien, in his paper, has stated that “it allows for several different research tools to be used as the project is conducted”, and “various methods, which are generally common to the qualitative research paradigm” (O’Brien, 1998).

While working on the action research, several unstructured interviews will be made with Finnish enterprises who have running web application developed in framework and colleagues who have been working with frameworks to gain more opinions from different points of view. Besides, published sources and reports, papers will be used as secondary sources.

1.5 Scope and limitations

The first and foremost, the scope of thesis is limited to web application only. The thesis introduces Zend Framework, a PHP framework. Although some other PHP

frameworks will be studied besides Zend Framework in order to compare the differences between them, these studies only give the general ideas but don't dig deep in other frameworks than Zend.

As this study is based on Tykes project using action research methodology, only relevant features related to the project are considered: learning curve, database management (including database configuration, mapping, and predefined methods), user management, internationalization, and security.

In this thesis, using framework as a method will be compared against other coding methods in order to point out advantages and disadvantages of the framework. The other methods include coding by hand, using Content Management System (CMS) or What You See Is What You Get (WYSIWYG) method. Throughout the study, coding by hand has proven its strength and flexibility above the other two methods. Moreover, the characteristics of coding in framework and coding by hand are more comparable. Therefore, this study will concentrate more in comparing the two mentioned methods. CMS and WYSIWYG will also be study and introduced in general view though.

As mentioned above, Web application frameworks could be easily mistaken with other type of web toolkits. A good comparison will be carried out in the latter part of this study in order to make a clear border between those.

1.6 Thesis structure

In Chapter 1, we have talked about the research structure, and strategy, as well as its scope and limitation. This thesis targets to novice web developers seeking for tools to start their careers. Therefore, we are starting with the theoretical part, explaining each and every techniques and tools included and why should we end up with the current tool. Chapter 2 introduces the web framework, talking more about the different and the nature of framework for web application. There will be some comparisons between web framework and other popular development techniques, such as using Library, CMS or WYSIWYG. The last part of Chapter 2 will talk about the main design pattern which is usually used in most web framework that is MVC. This chapter also includes example codes to illustrate the

design difference between coding PHP by hand and code using frameworks. In Chapter 3, we go on with the languages. A comparison chart between languages is presented, especially between those designed for developing web application. Chapter 3 also introduces some popular framework for PHP, which is the programming language used in this thesis. These frameworks are arguably the most popular PHP frameworks, based on specific criteria that will be mentioned beforehand. In Chapter 4, main features of a web application are compared between a code with and a code without framework. Codes without framework include codes done manually by hand, with CMS or WYSIWYG. Note that not all methods may be compared in the same feature, mostly between manually coding and framework coding. At the end of Chapter 4 will be a table to sum up the comparisons. Finally, Chapter 5 presents the conclusion of this thesis, which includes own opinions after the research about the chosen tool, its advantages and disadvantages and suggestion for further works.

2 INTRODUCING WEB FRAMEWORK

2.1 Web Framework definition and characteristics

There are different kinds of frameworks which support different tasks. However, they all have the same requirements and do the same tasks. They make it easier to work with complex technologies and promote consistent coding, fewer bugs and provide more complex application (Clifton, 2003).

So, what is exactly a web framework? In general, a web framework is considered a skeleton of an application into which developers plug in their code and provides most of the common functionality. It is built from a collection of pertinent objects, factored into classes, so that the framework can be reused, either its design or code (Gamma, Helm, Johnson, & Vlissides, 1995).

In the same way, web application framework is defined as a collection of source codes organized into certain architecture that can be used to support the development of dynamic websites, web applications and web services. By that mean, developers can think of framework as half-done applications that can be built upon to extend them to complete a particular application by creating application-specific subclasses of abstract subclasses from the framework. In fact, any half-done applications could be released and called themselves “framework”. They are considered bad frameworks that damage the reputation of the good ones.

Let’s take a look at an example code, written in PHP, to address the current issues of web pages written in pure PHP without the help from any framework.

```

<?php
$link = mysql_connect('localhost', SQL_USER, SQL_PASS);
if(!$link) :
    die('Could not connect: ' . mysql_error());
endif;
$db = mysql_select_db(SQL_DB, $link);
if(!$db) :
    die ('Can\'t connect to database : ' . mysql_error());
endif;
?>
<html>
<head>
<title>PHP Example</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>
<body>
<?php
$result = mysql_query("SELECT * FROM users");
while ($row = mysql_fetch_array($result)){
    $fullname = $row['fullname'];
    echo "<ul> <li>$fullname</li> </ul>"
}
?>
</body>
</html>

```

FIGURE 1: Example code in PHP

First, the code opens a connection to a database to run and retrieve the names of all current users. Looping over the user list, it generates an HTML list of the names.

This is a very simple one-of page, even a novice developer with little knowledge can understand everything from start to end. It is also easy to deploy: saving it to a “.php” file to any server and visit the page with any web browser. With it, the write-from-scratch approach isn't a very bad idea. However, because of its simplicity, there is nothing to learn. As the result, 3 main issues are rising which can affect any projects.

First of all, even a simple web application usually contains more than 1 page. So, what happen should multiple part of an application need the same function, say connection to a database in the above example? It is surely no need to duplicate

the fraction of code into each script. One simple solution for most developer would be to refactor it into a public function or using namespace.

Secondly, it isn't a good practice when a PHP developer has to take care of all HTML tags and other metadata. This may reduce productivity level of the developer and increase chances of mistakes. Moreover, separating logic part and the HTML part is a good practice as it allows a coder and a designer working simultaneously.

Finally, with the above mentioned separation, a designer who has no experience in the code language will be able to redesign the page without any chances which may crash the whole application. In other words, the designer may edit the page appearance without affecting the logical part.

Precisely, these issues are the main reasons why developers should work with frameworks as those are what web frameworks mean to solve.

2.2 Web Framework in comparison

2.2.1 Web Framework against Library

Conventional subroutine library, or sometimes called toolkit for the same matter, is a set of prebuilt modules which can be taken into use in applications. In many cases, several libraries could be used in one single application.

The main difference between Library and Web application framework is the level where they can be implemented in. When using a library, developers write the main body of the application then call the desire codes of the library. It doesn't invoke developers' code. Framework, on the hand, acts as the main body which calls the codes written by developers. In this way, developers have to write specific operations and calling conventions. However, it may reduce the decision making in structure design. This phenomenon is Inversion of Control, because the data flows in an inversion way compare to the ordinary course of programming. It is also known as "Hollywood's Law: Don't call us, we'll call you" (Sweet, 1985).

A comparison between a command line program and a window program is a simple but clear example of inversion of control. In a command line enquiry, developer writes a code that controls the system, deciding what to ask or when to ask the user. When using a window program, the window system is in control based on the binding the developer made when creating the form. It decides when to call the developer's method.

Inversion of control is a common phenomenon occurring regularly when extending framework, makes framework clearly different from library (Fowler, 2005). The difference here is quite obvious. They behave differently. However, some developers think of framework as something bigger and more complete than library. The word "framework", eventually, becomes more popular and overused. That is why some large libraries are sometimes confused as frameworks.

2.2.2 Framework against CMS and WYSIWYG

CMS and WYSIWYG are the other 2 popular methods in development. Similar to framework, they support in building applications. CMS stands for Content Management System while WYSIWYG stands for What You See Is What You Get.

A content management system is software that keeps track of every piece of content on your Website, much like your local public library keeps track of books and stores them (Joomla). A CMS platform includes a large amount of add-on modules and designs, letting any user with little to non-coding knowledge to build a website fast but still quite flexible.

On the other side, Microsoft FrontPage and Adobe Dreamweaver are the 2 popular examples of WYSIWYG. In general, WYSIWYG is a HTML editor that hides all HTML code from the user. It displays exactly how the site will appear on the display screen and allows users to make changes directly from there, making it possible for non-technical users to make changes to an existing website or create a new website easily.

In general, framework, CMS and WYSIWYG have the same purpose, which is support the production of web applications. With CMS and WYSIWYG, users can process with least knowledge of coding. Framework requires a fair level of coding knowledge and is considered a tool to support professional web programmers.

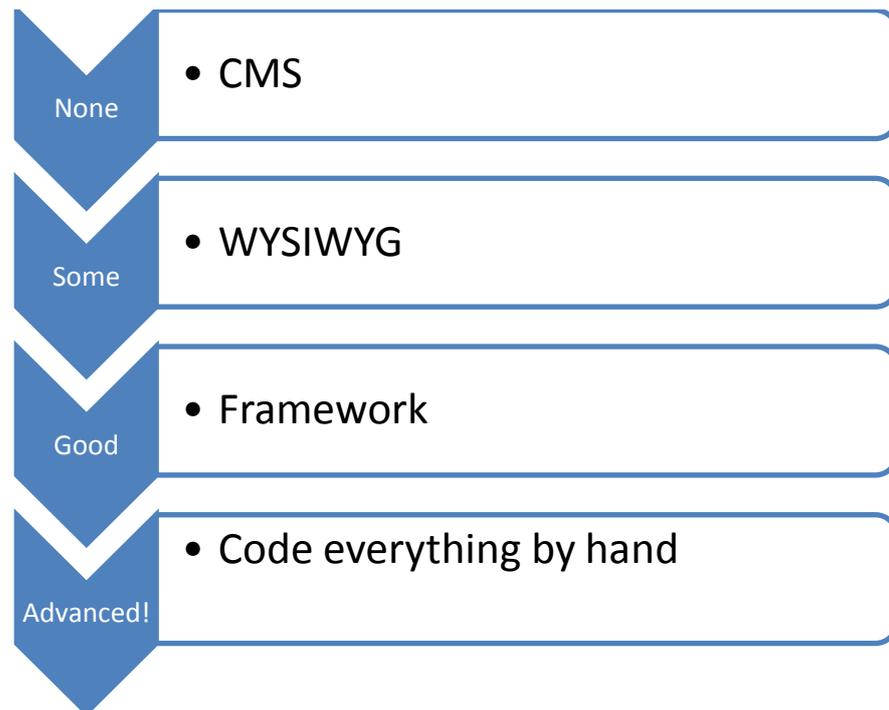


FIGURE 2: Technical knowledge needed (Bergeret, 2010)

2.3 Design pattern of Web Framework

In object-oriented programming, design pattern is defined as a general solution to a commonly occurring problem (Marston, 2007). Nowadays, many frameworks take advantage of design pattern to ease the development process. MVC becomes the backbone of most frameworks.

First introduced in 1979, MVC was used to build user interface in Smalltalk-80. Since then, the model has become the most well known design pattern because of its ability to increase flexibility and reuse. It consists of three kinds of objects.

Model: is the application object. It represents the data structure with all relationship and dependencies and provides an interface to manipulate all classes that correspond to logic object of the application.

View: is the screen presentation. A View should never modifies the application data, but just only present it. An application may have multiple Views for the same data to be display in different devices.

Controller: is the glue between Model and View. It handles user inputs and defines the way user interface react to them but should only use the methods provided by the Model.

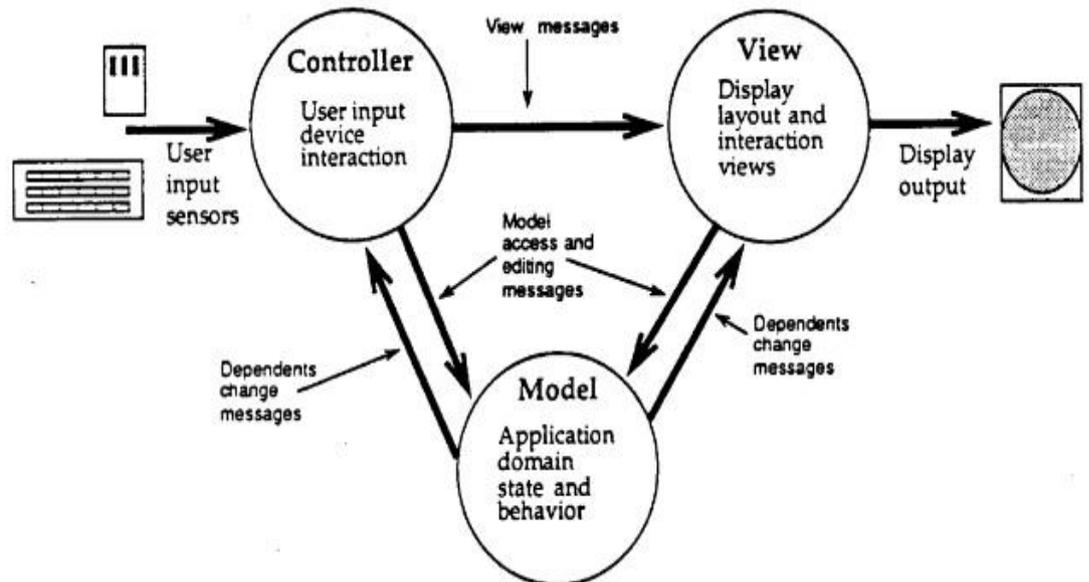


FIGURE 3: MVC model in SmallTalk80 (Krasner & Pope, 1988)

Below is an example of MVC architecture, written in Zend Framework.

```
resources.db.adapter      = "PDO_MYSQL"
resources.db.params.host  = "localhost"
resources.db.params.username = "YOUR_DB_USERNAME"
resources.db.params.password = "YOUR_DB_PASSWORD"
resources.db.params.dbname = "YOUR_DB_NAME"
resources.db.params.charset= "YOUR_DB_CHARSET"
```

FIGURE 4: Define database link in Zend Framework

```

class UserController extends Zend_Controller_Action
{
    public function indexAction
    {
        $user = new Default_Model_UsersMapper();
        $this->view->entries = $user->fetchAll();
    }
}

```

FIGURE 5: Controller_The application logic

```

Username: <br />
<dl>
    <?php foreach ($this->entries as $entry): ?>
    <dt><?php echo $this->escape($entry->username) ?></dt>
    <?php endforeach ?>
</dl>

```

FIGURE 6: View_The application presenter

```

class Default_Model_DbTable_Users extends Zend_Db_Table_Abstract
{
    /** Table name */
    protected $_name = 'users';
}

```

FIGURE 7: MySQL database table

```

class Default_Model_UsersMapper
{
    protected $_dbTable;
    public function setDbTable($dbTable)
    {
        if (is_string($dbTable)) {
            $dbTable = new $dbTable();
        }
        if (!$dbTable instanceof Zend_Db_Table_Abstract) {
            throw new Exception('Invalid table data gateway provided');
        }
        $this->_dbTable = $dbTable;
        return $this;
    }
}

```

```

public function getDbTable()
{
    if (null === $this->_dbTable) {
        $this->setDbTable('Default_Model_DbTable_Users');
    }
    return $this->_dbTable;
}
public function fetchAll()
{
    $resultSet = $this->getDbTable()->fetchAll();
    $entries = array();
    foreach ($resultSet as $row) {
        $entry = new Default_Model_Users();
        $entry->setuserID($row->userID)
            ->setUsername($row->username)
            ->setPassword($row->password)
            ->setUserType($row->usertype);
        $entries[] = $entry;
    }
    return $entries;
}
}

```

FIGURE 8: Model's methods used in application controller

```

class Default_Model_Users {
    protected $_username;
    public function __construct(array $options = null)
    {
        if (is_array($options)) {
            $this->setOptions($options);
        }
    }
    public function __set($name, $value)
    {
        $method = 'set' . $name;
        if (('mapper' == $name) || !method_exists($this, $method)) {
            throw new Exception('Invalid user property');
        }
        $this->$method($value);
    }
    public function __get($name)
    {
        $method = 'get' . $name;
        if (('mapper' == $name) || !method_exists($this, $method)) {
            throw new Exception('Invalid user property');
        }
        return $this->$method();
    }
}

```

```

    }
    public function setOptions(array $options)
    {
        $methods = get_class_methods($this);
        foreach ($options as $key => $value) {
            $method = 'set' . ucfirst($key);
            if (in_array($method, $methods)) {
                $this->$method($value);
            }
        }
        return $this;
    }
    public function setUsername($username)
    {
        $this->_username = (string) $username;
        return $this;
    }
    public function getUsername()
    {
        return $this->_username;
    }
}

```

FIGURE 9: MySQL table's structure

Although the code in framework appears to be longer than the original, it still could be shortened by not defining methods in class `Default_Model_UsersMapper` but using SQL query as a string instead. However, using SQL query is considered a bad practice in framework. Moreover, predefined methods in model mapper can be called easier any time when needed, thus prevent duplication of the SQL query.

The above code is the standard of a framework, including database definition, model definition, controller, and view. More of them could be forms, and page layout, for the whole page where data appears to be the same or little changes.

Besides MVC, there're also many other popular design patterns commonly used in web framework such as Singleton, Prototype, Factory Method and Decorator. However, in the scope of this study, MVC is the main design pattern used in the studied web frameworks. Full description about design patterns can be found in an influential book of the Gang of Four, namely *Design Patterns: Elements of Reusable Object-Oriented Software*. In fact, MVC also uses other design pattern, Decorator to add scrolling to a view for example.

3 WEB APPLICATION LANGUAGES

Which server-side platform is better may be the most common question when starting a new web application project. This is not an easy question and unfortunately, there is no instant tool for it. In general, there are two ways to a decision. The common way to make decision, usually applied by junior developers, is simply using the language they know the most. This is not a good way in professional work. However, it minimized the risks working on a less familiar language.

The other way to use when discuss this issue is following an activity diagram by answering a list of architectural choices. The chart bellow shows an example of the decision making process. Following this way, every project should be approach individually and each choice made must be based on the project's requirements, not based on any developer's strength (Synodinos, 2007).

In general, language is just a tool for a developer to work with. It isn't really matter which programming language using, as long as the customer's requirements is acquire on time and within budget. However, it matters as we all are familiar and get passionate about our tools.

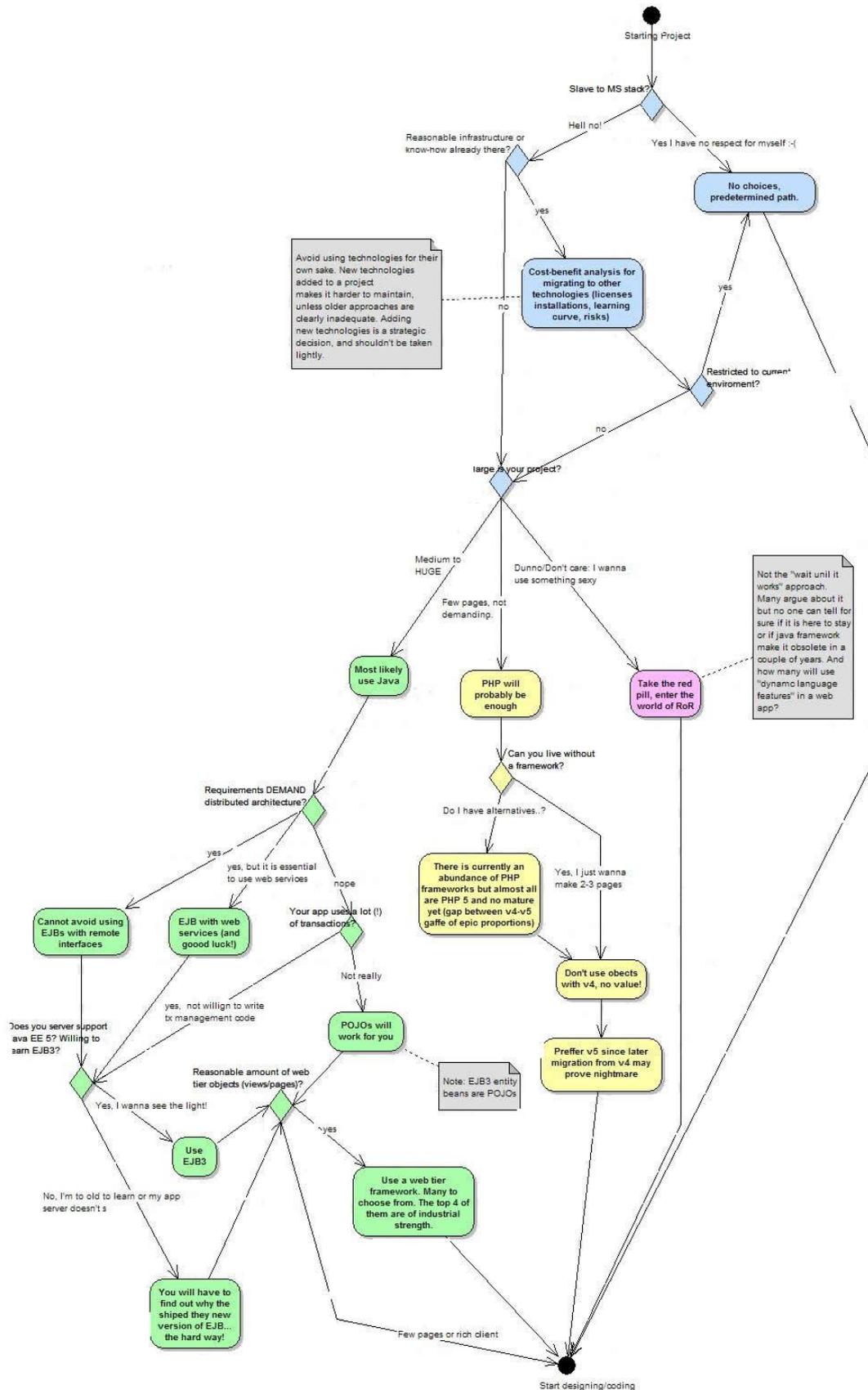


FIGURE 10: Diagram for decision making (Synodinos, 2007)

3.1 PHP among other languages

Which language is the most popular one in the web-world? There are several ways to measure the popularity of a language. For example, it can be based on the number of:

- Available jobs required the language's skill
- Searches on every search engines
- New/existing application written in the language
- Developers using the language

Using different ways, a number of surveys have been attempted to rank programming languages. Tiobe is specialized in assessing and tracking the quality of software. It's ranking of programming languages is one of the most popular and reliable in the business. The rating is not about the best programming language but about the popularity of the languages based on the number of skilled engineers world-wide, courses and third party vendors (Tiobe programming community index, 2012).

TABLE 1: Programming Language position in 1996, 2006 and 2011 (Tiobe programming community index, 2012)

Programming Language	Position In Dec 2011	Position In Dec 2006	Position In Dec 1996
Java	1	1	5
C	2	2	1
C++	3	3	2
C#	4	8	-
Objective-C	5	42	-
Php	6	5	-
(Visual) Basic	7	4	3
Python	8	7	26
Perl	9	6	6
Javascript	10	10	25
Lisp	13	17	16

To see a bigger picture between some of the most popular language nowadays, the table above shows the position of the current top popular language compared to 5

years ago. In the mention period of time, Java has been a leading language and C remained in the second position. However, these languages are hardly implemented in web application because their implementation in low level making them even more complicated. Therefore, between the main web languages, namely PHP, Python, Perl and Javascript, PHP has the highest rank and probably the most anticipated web application language at the moment.

Compared to Python, which had its first implementation during December of 1989 and currently is the second best web languages according to the above ranking, PHP is a few years younger. Originally named “Personal Homepage Tools”, PHP was only a set of Perl script. Its development began in 1994 and official launch came in June 1998. Although coming later, PHP had an impressive debut and already came to fifth place in the ranking in December 2006. In 2004, it was chosen as a “language of the year”. A few years later, Python made it twice, in 2007 and 2010. PHP is still the most popular web language at the moment, based on its usage. However, Python has made a great breakthrough proving its easy and efficient implementation successfully.

3.2 Introduce some popular PHP framework

There are several factors when looking to a framework:

- Learning curve
To start developing with a framework, a developer has to go through 2 learning curves instead of 1, the first one for the language and the second for the framework. Although the application is written in PHP, a framework still has its own language infrastructure. Therefore there’s need of learning the framework’s terminology. It makes the learning curve sometimes very steep.
- Documentation
A good documentation includes:
 - User manual
 - Example codes and snippets

- Activity level of the user community
A good documentation is a must but not enough without a good user community. When most of the frameworks are still evolving, the documentation might be patchy sometimes. Moreover, other informal tutorials, and comments would be good to evaluate a framework.
- Frequent updates and bug fixes
This would mean more functions come in but must not cause software unstable.
- Performance
Performance might be an important factor should be considered when choosing a framework for a project but it alone is not the whole story, otherwise we should use plain HTML or PHP instead. It' about how a framework performs its feature set with such a performance.
- Work market available
A significant factor especially for novice developers choosing the framework for the upcoming work life is the amount of jobs available in need for that framework.

3.2.1 Zend Framework

Zend Framework is an open source, object oriented web application framework for PHP5 (Zend Framework Quickstart). The framework project is sponsored by Zend Technology Ltd., the company that develops the PHP language. Zend has a great support which makes the Zend Framework becomes one of the most popular PHP framework of all.

Zend Framework is designed to be simple and the component is loosely coupled. For some points, the framework is considered more as a library of components, which can be used when needed. Unlike other popular framework such as CakePHP and Yii, the MVC architecture is optional. This option lowers the learning curve for novice as well as increases the flexibility of the framework for experience developers.

At the time writing this study, Zend Framework 2.0 beta1 has been released, the first in a series of beta released cycle. The Zend documentation is excellent for the current stable version 1.11. It's thorough and includes all features and libraries of the framework. The user community activity is quite moderate, though couldn't tell exactly because the archives doesn't show dates.

Because of the ability to scale up much more, Zend is the first choice for many large projects. For small and medium projects, however, using Zend would make them even more complicated to develop because the framework itself is quite large.

According to the Monster UK site, beyond the 3 selected frameworks, Zend has the highest hit rank with 129 hits on 20th of January 2012. In the graph below, Indeed.com makes a clear comparison on jobs posted for the frameworks.



FIGURE 11: Job Trends for Zend, CakePHP and Yii (Job trends, 2012)

3.2.2 CakePHP

CakePHP is a rapid development framework for PHP that provides an extensible architecture for developing, maintaining, and deploying applications. With over 13 million hits on Google search (Google, 2012), CakePHP is arguably the most popular PHP framework at the moment.

The most important factor that makes CakePHP becomes a popular PHP framework is its well structure and thorough documentation. Although CakePHP has a unique terminology set, making it hard for new developer to approach, its documentation offers a good help to reduce the learning curve for new incoming developer to pass through. There are books for every present version. Even the new 2.0 released candidate has an in-development book to present the new features. The community is very helpful and the Q&A answer is well built with many thorough answers for each question in different sections. For case by case help, a massive online community is there to provide it. Compared to all kind of open-source programming group on the Web, the PHP community is the largest.

At some points, there are still some drawbacks of CakePHP. The first thing is its support of PHP4, which limit its features from more advance features of PHP5, magic overloading functions to name one. Backward compatible is good, but since PHP5 has been around long enough to mature and most of share hosting supports it out of the box, there's no place for PHP4 anymore.

Second, its strict rules at some points are good to build good programming practice, but preventing flexibility isn't a small drawback. As mention in Zend, CakePHP strictly implements MVC model while it's an option in Zend. In Yii, convention is a favor but CakePHP enforces it.

3.2.3 Yii

With a performance test under its own web page, Yii claims to be the fastest framework among CakePHP, Zend and Symfony by implementing lazy loading technique to acquire an astonishing performance. Using this technique, a class or an object won't be included until is accessed for the first time so the application requires less resource to perform. However, the test has been performed long ago so the result may somehow less accurate because most of the mentioned framework has been announcing several newer versions with bug fixes and performance enhanced together with new features. The performance of the frameworks, moreover, can be easily tuned with proper cache configuration and as long as the ActiveRecord pattern is used carefully, it isn't that far apart.

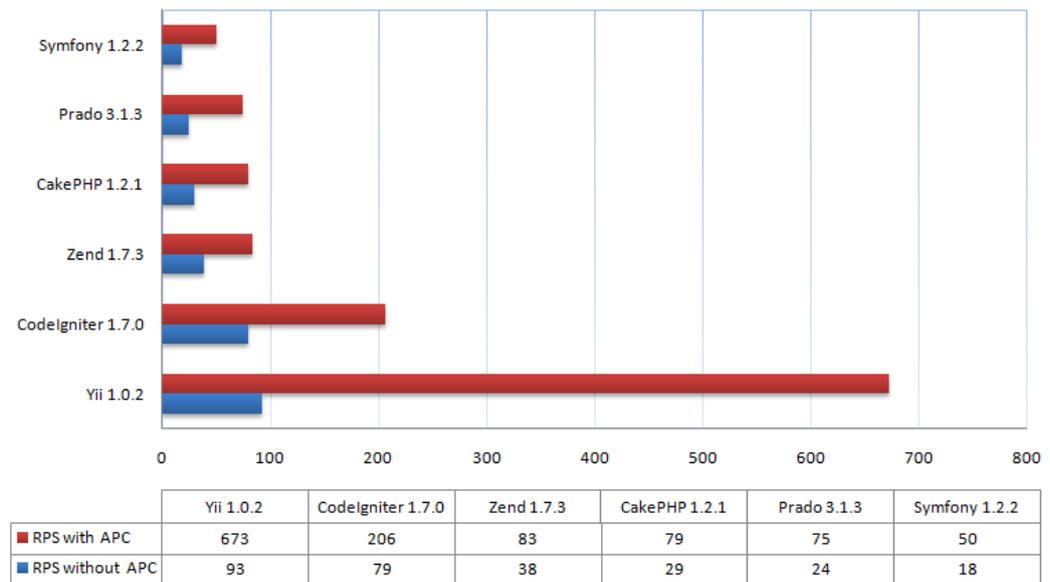


FIGURE 12: PHP frameworks performance in comparison (Framework performance comparison)

3.3 Comparison between the frameworks

Each framework has its own advantages and disadvantages most important in performance, documentation and user community. To start working with a web application project, choosing a framework may prove critical. A complex framework is necessary for large projects but may consume more time than simpler framework when apply for small and medium projects.

To come up to the decision of choosing Zend as a tool for research in this thesis, I've done a few interviews with some small Finland entrepreneurs that have web application in development process and opinions of professor Alix Bergeret from Wolverhampton University, The United Kingdom. Considering the above criteria plus a poll opened on an open-source forum with question "Which PHP framework you prefer to use for your future web application projects?" (Which PHP framework holds a promise for the future, 2009), Zend is a promising framework and a good choice for many projects because of its ability to scale. Moreover, the current version is very stable. Though, the beta releases may

add more features and bug fixes in near future. Last but not least, it's backed by the leading PHP company which makes it more reliable.

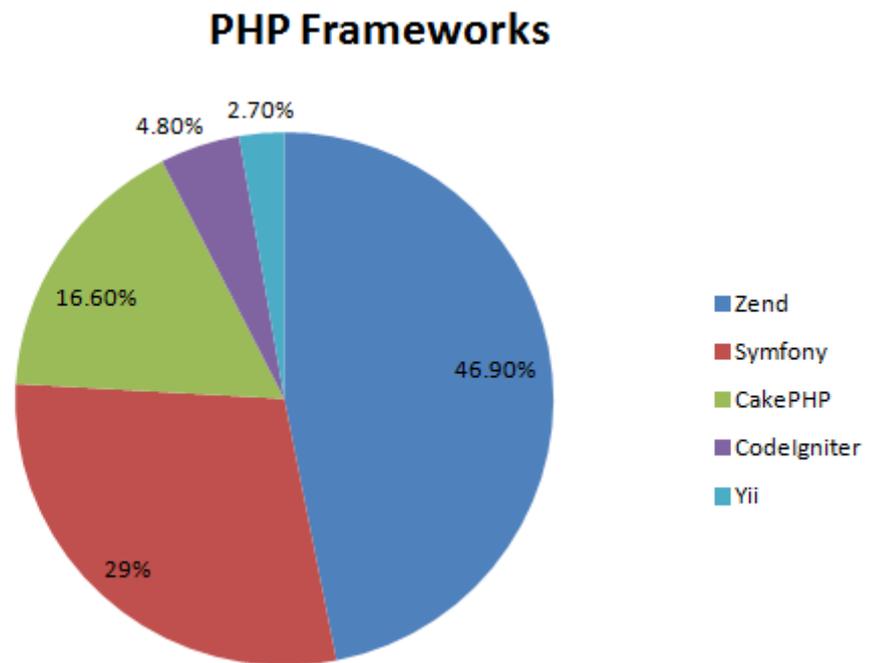


FIGURE 13: Result for "Which framework is the most promising?"

4 FEATURES COMPARISON

4.1 Model View Controller

Since described in SmallTalk 80, MVC has been the widest used programming pattern. In reality, experienced developers tend to separate the user view and the business logic by putting them in separate files and include them when needed. However, this approach may be good for small to medium size project, it's getting much more complicated to use in large project than applying MVC. Sometimes, it's even mistakenly realized as MVC because MVC is so popular and frankly, some parts of the classic MVC doesn't really make sense in current rich clients.

Act as the backbone of an application, MVC is one of the most important factor to be considered in application design. At the heart of the MVC is the idea of Separated Presentation. MVC tends to make a strong separation between presentation (the view and controller) and domain (the model). Domain objects, therefore, should be completely confined to it and they should support multiple presentations simultaneously.

With framework, MVC is applied with ease. Some frameworks apply strict rules on using MVC. Others make it available as an option. When coding by hand PHP, applying MVC requires a very good knowledge in the model pattern. It requires a significant planning hence a deeper level of complexity that requires a considerable attention to smaller details. For this reason, MVC is usually overlooked by developers for small projects and sometimes even for medium size projects because the extra effort may not worth it.

In the figure below, it describe the data flow in a typical PHP program without using any framework and MVC. In the figure, the PHP script serves like a bridge between client and database. It handles everything, from handling user input to manipulating database and presenting the result. To reduce redundancy, using the include() function is the most common way in which developers take out common objects to store into external files.

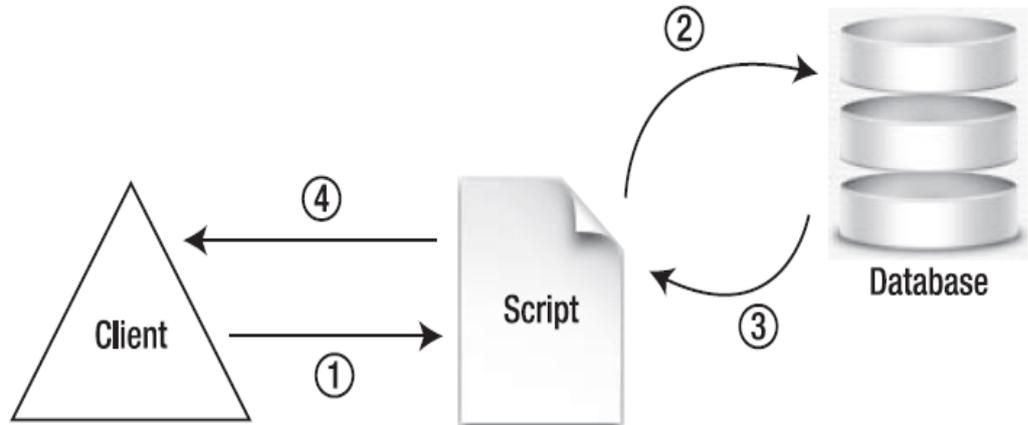


FIGURE 14: Typical Flow in PHP Programs (Golding, 2008)

MVC improves the PHP flow by effectively dividing the flow into smaller steps and separating them clearly, thus making sure that each object is written once and only once. The flowchart below illustrates the application flow in an application written with the CodeIgniter framework.

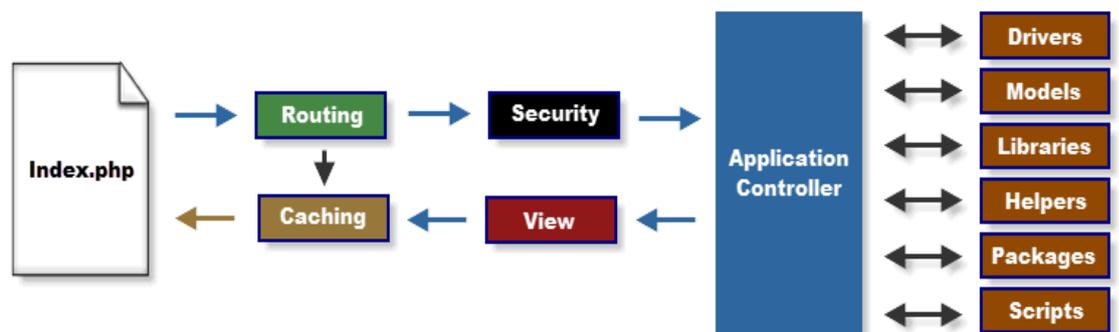


FIGURE 15: Data Flow in CodeIgniter Programs (CodeIgniter User Guide, 2012)

In the flowchart above, the Index.php serves as the front controller. It's the main controller initializing the resources needed to run the application and is the first component to run when the application starts. When receiving a HTTP request, the router examines and decides what should be done to respond. If a cache for the request exists, it is sent directly to the web browser. If not, the request and any data submitted will be examined for security before an application controller is loaded to handling the request. The controller loads any needed resources to

process the specific request and the finalized view will be rendered to the web browser for user.

4.2 Learning curve

In the Figure 2: Technical knowledge needed for different approach, CMS and WYSIWYG require significant less coding knowledge than using framework or code by hand because they aims to ease the process to the lowest possible level. By doing that, it sacrifices the flexibility and limits the functions of the application. Most current CMS and WYSIWYG development application doesn't offer a way to reduce redundancy of code as its top prior feature is to make development process easier thus it also sacrifices the application's performance. As the result, these approach usually suitable for non developer as they could only be used for small project such as personal site.

Framework as well as coding by hand, though require a phenomenal amount of knowledge more, are the approaches for professional developers. It offers more options for developer to choose, such as between MVC and being more flexible in program structure, and can be used for all kinds of project. However, when talking about learning curve, also in Figure 2, there is a small difference between the two approaches. Although novices may find themselves hard to learn a new language plus any terminologies of a particular framework, using framework require a relative lower level of coding language and less experience than coding PHP by hand.

However, there're still some great distinctions between the two kinds, code in framework and by hand. A great argument has been going on whether PHP is good to use on large websites. Notably, a "large" website is just an indefinable measurement mechanism. To make a decision, several questions should be considered. They are:

- How much control on the deployment platform will you have?
- How many numbers of distinct pages will the site have?
- How complex will the site be upon deployment?
- How large is the team working on the site, now and in the future?
- How much experience do the developers have?

In contrast, using framework, developers can easily build large websites, or easy to scale up when needed with the help of MVC. Most framework claims itself for the sake of simplicity and productivity. The site is divided in to two parts, presentation and business logic. In this way, developer can make changes to either one without affecting the other, changing the business logic or update its presentation with ease.

A very good example in using PHP on very large website is Facebook, the largest social network at the moment. Although the backend server is written not only in PHP but also in many languages such as C++, Java, Python, Perl, Ruby, and others, at the frontend, their servers run on LAMP (Linux, Apache, MySQL, and PHP) stack. At EmTech@MIT 2010 conference (EmTech@MIT 2010, 2010) in September 2010, Mike Schroepfer did an interview concerning the scaling ability and performance of site built in PHP. Facebook has been growing so quickly since its startup that the database growing huge in both size and complexity. The Figure in Appendix 2 shows an example of Facebook's database. Due to unique characteristics of social networking, its database has an extraordinary amount of interconnected dataset. Facing these issues, Facebook started a phenomenal amount open source projects and backend services.

4.3 Database

For database management system, most web frameworks choose to use MySQL because it works well together with PHP as well as other languages to manage database for web application.

MySQL is obviously the most popular relational database management system (RDMS). It has several free and paid editions for developers to choose. Although

the free edition already is a full-featured database management system, paid editions add more choices to support developers in commercialized projects. Besides PHP, it works well with other applications such as Joomla, Drupal, and WordPress. MySQL is one of the core opponents of the widely used LAMP (Linux, Apache, MySQL, and PHP) web application software stack, which is used by Facebook for its front-end server.

Although NoSQL has been introduced since around 2009 by those who aren't pleased with MySQL, claiming that NoSQL has even stronger consistency and better transaction integrity, MySQL is still a preferred choice for web developers because of its easiness to work in web application environment and much larger support community. Moreover, because of the lacks in availability and scalability, many huge sites such as Google or Amazon chose to live with MySQL, sacrificing consistency for availability and scalability. However, as NoSQL continues to rise, Google has added NoSQL support in Google Web Toolkit, a well known framework-like web application development kit provided by Google.

4.3.1 Connection

In order to use database, a connection must be established between the application and the database server. The connection requires credentials and should be placed separated in a restricted access area is a good practice. However, this is rather hard and requires a good knowledge of PHP when coding by hand. The Figure 4 in Chapter 2 shows how the connection is established in Zend Framework. The connection is defined and kept in a separate file where outsider can't access to keep the credentials safe. This is done with ease in most frameworks. In the figure below, a simple but bad practice usually is done by developers coding PHP by hand. In the code, username and password of the user is shown publicly and could be easily exploited by hackers.

One way that has been used widely and has been mentioned above is using `require()` or `include()` function, but this may cause the code becomes much more complicated even in medium size site.

```

$db = new mysqli ('localhost', 'db_username', 'db_password', 'db_name');
if (mysqli_connect_error())
{
    echo "Error: Could not connect to the database.";
    exit;
}
// Working with the database then exit upon finish
$db->close();

```

FIGURE 16: Database connection in PHP without framework

4.3.2 Mapping and Methods

```

//input data getting from the form to the database
$query = "insert into table_name value
        ('".$_POST['username']."', '".$_POST['password']."')";
$result = $db->query($query);
if ($result)
{
    echo "New username inserted to the database.";
} else {
    echo "An error has occur.";
}

```

FIGURE 17: Data input in PHP without framework

The code above can be used to work with the value entered by user through forms. Although the code is quite simple for the sake of simplicity in this study, real project work would require much more effort.

In framework applying MVC pattern, besides separating the database connection credentials, data models and functions may also be separated just like an example in Figure 4 in Chapter 2. In the Figure 5, data submitted by user through the form is handled and save to the database in controller. Because the save() function has been pre-defined in the model mapper, it can be called easily by the controller. This practice may also reduce code duplication as pre-defined functions can be called whenever needed.

4.4 Security

Cross-site request forgery (CSRF) is a very common vulnerability in any web application with forms. In general, a member with established level of privilege with the target site is a victim and also an unknowing accomplice. In CSRF, a victim is the one who actually do the attack, not the attacker, which makes it even more difficult for web authority system to detect and prevent the attack. It can easily bypass usual authority mechanism including authentication, identification, and authorization and can only be mitigated by a specific form validation, mostly by using token.

In a typical scenario, the attacker tries to trick the victim to load a page containing a malicious request, which is able to inherit the victim's identity and privileges to perform unwanted actions on the target site under the victim's behalf. Therefore, the site will not be able distinguish the attack from a legitimate user request. If the victim is the administrator account, the attacker may compromise the whole application.

As the result, the only way to prevent a CSRF attack is in the form design by using an anti-CSRF generated token. The token is generated before the form is loaded and stored in session. When user submits the form, the token is checked before the form is processed. If the token is mismatched, the form will be terminated immediately.

In Zend Framework, thanks to the request/response mechanism, the token validating process is pushed up fast and easy. Within a one-line code in the form, the application is protected against CSRF attack.

```
$this->addElement('hash', 'csrf', array( 'ignore' => true ));
```

FIGURE 18: CSRF hash implementation in Zend

With no other action required, when a csrf hash token is added in the form, it will be automatically checked upon submit. Then, the form is processed normally by the controller.

```

public function exampleAction() {
    $request = $this->getRequest();
    $form = new Default_Form_Example();

    if ($this->getRequest()->isPost()) {
        if ($form->isValid($request->getPost())) {
            $model = new Default_Model_Example($form->getValues());
            $model->save();
            return $this->_helper->redirector('index');
        }
    }
    $this->view->form = $form;
}

```

FIGURE 19: Form controller in Zend

In coding by hand, the token can also be generated easily by using md5() function, for example. Moreover, the token, as a session variable, should be initialized.

```

if (!isset($_SESSION['token'])) {
    $_SESSION['token'] = md5(uniqid(rand(), TRUE));
}

```

FIGURE 20: CSRF has implementation without framework

The token is inputted hidden inside the form, which will be validated when the form is submitted.

```

if ($_POST['token'] == $_SESSION['token'])
{
    /* Valid Token. Do something */
}

```

FIGURE 21: Form validation without framework

A csrf attack is very dangerous because the request comes from the victim, which makes it possible to attack applications that only the victim can access. With just small steps in form design, the thread is eliminated. However, every form in a single application is vulnerable. To totally eliminate it, tokens should be implemented in each and single form.

4.5 Internationalization and localization

As the www spreads, web application gains more and more usage. One significant advantage of web application is unrestricted to geography that users can easily access wherever they are. However, one rising problem is that users more likely to use applications in their mother tongue than other second languages. Although you can buy in any language, you must use the customer's language to sell (Morgan, Luttrell, & Liu, 2001).

Thus, internationalization (i18n) and localization (l10n) are important features for applications available in more than one country using different languages. I18n and l10n are two parts of the globalizing an application and both are orthogonal to the application layers.

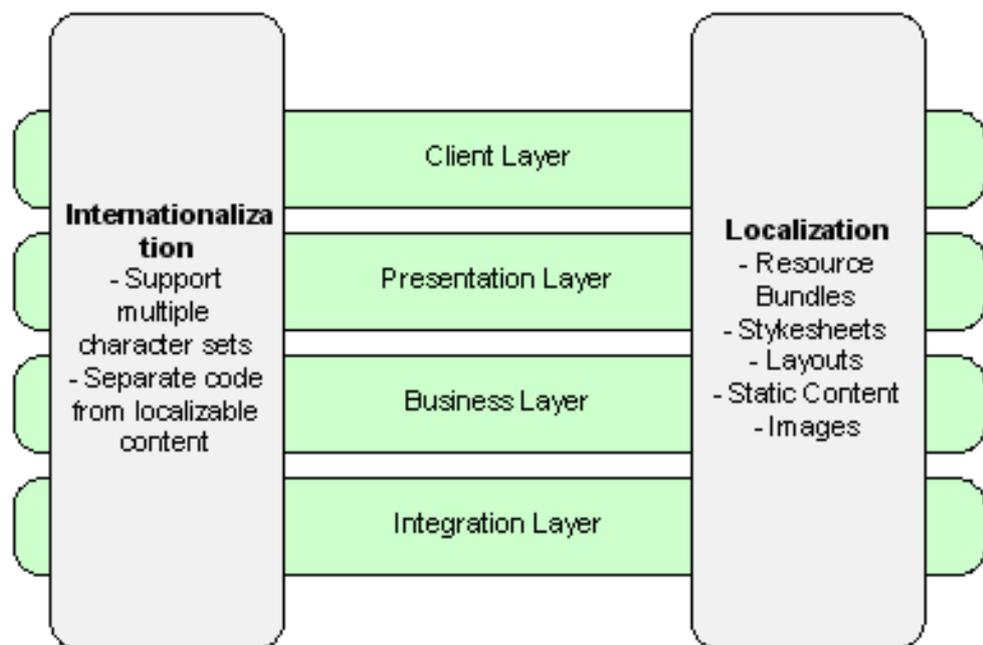


FIGURE 22: Internationalization and localization are orthogonal to the application layers (Sachdev, 2007)

In general, PHP itself only support native array and gettext. Gettext is an open-source project which can be used to internationalize PHP applications. However, the usage of gettext is very complicated. At first, every string which is shown in the view must be wrapped in the gettext function. The translator will create a new .po file from the template which is filled by the translation and then be edited

either by hand or by third-party translator tools. Finally, these files will be compiled and ready for distribution. When running, the program will display the text in selected language if the file is available for it.

Besides using native array, gettext is considered the most comprehensive tool that supports i18n. However, as described above, the process is rather complicated and required using command lines and other third-party tools. Moreover, source code must be modified for gettext to work.

Zend Framework, on the other hand, has its own Zend_Translate package to support i18n and l10n. The package contains several adapters for developer to choose which one to use. Some of them are Zend_Translate_Adapter_Array, Zend_Translate_Adapter_Ini, Zend_Translate_Adapter_Gettext, etc. More important, Zend_Translate also allows self written adapter classes. Upon integrated, these adapter can be used just liked native adapter which already included in the package.

In both approaches, the method is used just for simple text strings and static data which appear in the application. Dynamic web applications usually use database to manipulate users' data. Thus, implementing i18n needs to works with database. In most cases, developer needs to create a new data table for each language and setup inter-connection between tables of different languages. In this case, web frameworks' ability in working with database proves helpful in i18n.

4.6 User management

User management might be one the most important feature in every application. Least privilege is an important feature in information security. The system should assign the least possible privilege for its users. In any case, granting more than needed permissions to a user can allow that user to obtain or change unwanted data or even damage the whole system.

A full user management process includes user authentication, identification and authorization. A common method of user authenticate is by using a database table to store user identity. The pair of user ID and password stored in the database will

be checked to see if the submitted combination is matched. After a successful authentication, authorization ensures the user can only work with its limited rights. The limited rights are granted by the application's admin followed the principal of least privilege strictly.

Due to its important, PHP takes a good approach in user management. The authentication process is very clear and straight-forward. User identification is sent via a form and matched with the corresponding pair from the database. The user session will begin after the user is granted access. The session will be destroyed when user decides to log out.

In Zend Framework, the process is very similar. However, with the help of `Zend_Auth_Adapter` and `Zend_Auth_Storage_Session`, user management is a little simpler yet more flexible. Moreover, usage of `Zend_Auth` eliminates the use of SQL queries making the code cleaner.

TABLE 2: Features comparison

	Code Manually	Code in framework
MVC pattern	<ul style="list-style-type: none"> • Hard to implement, leading to very complicated structure • Increase redundancy • The data flow is simple but not effective 	<ul style="list-style-type: none"> • MVC is available in many frameworks and easy to implement • Good for medium to large project but may lead to too complicated for small project • Redundancy is well avoided • Data flow is separated, leading to better performance
Learning curve	<ul style="list-style-type: none"> • Require the most knowledge in coding 	<ul style="list-style-type: none"> • Require less knowledge than coding but more than CMS and WYSIWYG • Extra effort to learn framework's unique terminologies
Scalability	<ul style="list-style-type: none"> • Very hard to scale 	<ul style="list-style-type: none"> • Enable easy scalability
Database	<ul style="list-style-type: none"> • Optional: database connection and mapping can be placed in different files and include() when needed • SQL queries are used 	<ul style="list-style-type: none"> • Database connection and mapping methods must be placed in a different directory • SQL queries and framework's adapter are two available options
Security	<ul style="list-style-type: none"> • The most effective CSRF prevention method is available, but more difficult to implement than in Zend Framework 	<ul style="list-style-type: none"> • The method is available and well supported by Zend Framework
I18n & L10n	<ul style="list-style-type: none"> • Can be implemented using third-party software • Two methods are supported: native array, and gettext • Native array is only suitable for static string • Using gettext is complicated 	<ul style="list-style-type: none"> • Methods are available with Zend_Adapter • Support wide range of demands • Gettext is available via Zend_Adapter • Self-written adapter classes can also be used
User management	<ul style="list-style-type: none"> • Good user management. • Native PHP methods are simple and effective 	<ul style="list-style-type: none"> • Good user management with support of Zend_Auth

To conclude this Chapter, features used in the two approaches are presented in the table above. In general, the use of MVC pattern and Zend_Adapter are two main elements which improve the general coding practice in Zend Framework. The controller in the pattern is the application's logic part which acts as a glue between the user's view and the application's model. Unique Zend_Adapter helps providing cleaner codes and eliminate SQL queries.

Native PHP language provides enough security measurement and a good user management. Zend Framework with Zend_Adapter again requires a bit less effort to gain the same purposes.

Last but not least, although coding PHP manually may require better knowledge of PHP language than in frameworks, novice developers may find it harder to learn coding in frameworks because of their unique terminologies. Moreover, implementing MVC pattern needs extra effort to learn.

5 CONCLUSIONS

There are many factors to consider when chosen a right tool for starting up. To answer the first question, we should analyze our needs and goals. This depends much on the work's requirements, and sometimes your current skills. First of all, developer should decide which language to use in a project. If it isn't set to a specific language, the best way is to stick to the language which you are most confident to use. An example of decision diagram was also presented. After that, to choose whether using any tool or coding by hand is also a critical choice. Although many companies have chosen web frameworks as a tool to aid its development process and improve coding standard because most of frameworks are very strict on this, it isn't necessary true in all cases. You may practically find it easier to code by hand while using framework may make it more difficult. The list of advantages and disadvantages of framework over coding by hand may prove helpful when making the decision.

Advantages:

- Easy working with dynamic content for project like social networking, online stores, news, and so on
- Easy scaling applications which can grow over time without needs in large changes in code
- Taking the most advantage of modularity and reusability of codes
- Increase ability to meet deadlines, rotating staff, and fitful customers which occur regularly in real-life development
- For any soon-to-be professional web developer, well equipped with web framework is an sufficient effort as it is valued well by employers

Disadvantages:

- Just purely an informative web page without any dynamic user-created contents such as personal portfolio
- Small projects with small database that requires limited database connection
- Such small projects wouldn't benefit much from framework's code

generation

- Some kinds of web application in which developers need to have total control over the code
- You or your teammates can't afford to learn how to use a framework

According to the above list of advantages and disadvantages, frameworks are created for common web applications but not all should apply. Some kinds of project such as extremely specialist websites, small and static websites are out of range. Since most of commercial web applications connects to a database and allow users to create and modify the content, frameworks grow quickly and developing with web frameworks becomes a common practice for professional developers.

If you decided to work with a framework, choosing which framework to work with is the next step. Again, this is affected by some other factors. Many of these factors have been presented in Chapter 3. Many of them such as:

- Learning curve
- Documentation
- Activity level of the user community
- Frequent updates and bug fixes
- Performance
- Work market available

Further works

At the time writing this thesis, many web application frameworks have matured enough for developer to work with confident. Most of them provides good stable versions and working on next versions with improved features and performance. However, there are still rooms for improvement. At the moment, the general problem occurring in development in all languages is that developer often needs to use more than one language in only one dynamic web application. Here is the current scenario:

- Code in X/HTML, or Javascript: for static web sites
- Code in PHP/Python/Ruby/etc: HTML code is called from inside which may become complicated and lengthy
- Templates are available in some frameworks, such as Zend, and Symfony. However, to some levels, they still require X/HTML.

As the result, a suggested further study would be an improved template built upon frameworks. One possible case is that the template is able to translate itself to X/HTML, Javascript, Ajax, and other essential languages needed to run on client side.

REFERENCE

2012. *Google*. [Online] 2012.

https://www.google.com/#hl=en&sugexp=efis&gs_nf=1&tok=CopI3uuWLu0MVUKPEMVDJQ&cp=7&gs_id=u&xhr=t&q=cakePHP&pf=p&output=search&scie nt=psy-ab&oq=cakePHP&aq=&aqi=&aql=&gs_l=&pbx=1&bav=on.2,or.r_gc.r_pw.r_qf .cf.osb&fp=144acd61eed23f45&biw=1366&bih=653.

Bergeret, Alix. 2010. *Web frameworks introduction*. Wolverhampton : s.n., 2010.

Berners-Lee, T. and Cailliau, R. 1990. *WorldWideWeb: Proposal for a HyperText Project*. November 12, 1990.

Burney. 2008. *Inductive & Deductive Research Approach*. 2008.

Clifton, Marc. 2003. What is a framework? *Code Project*. [Online] 11 3, 2003.

<http://www.codeproject.com/Articles/5381/What-Is-A-Framework>.

2012. CodeIgniter User Guide. *Code Igniter*. [Online] 2012.

http://codeigniter.com/user_guide/overview/appflow.html.

2010. EmTech@MIT 2010. *Technology review*. [Online] 2010.

<http://www.technologyreview.com/emtech/10/>.

Fowler, Martin. 2005. Inversion of Control. *Martin Fowler*. [Online] 6 26, 2005.

<http://martinfowler.com/bliki/InversionOfControl.html>.

Framework performance comparison. *Yii framework*. [Online]

<http://www.yiiframework.com/performance/>.

Gamma, Erich, et al. 1995. *Design Patterns: Elements of Reusable Object-Oriented Software*. s.l. : Addison-Wesley Professional, 1995.

Gilmore, Thomas, Krantz, Jim and Ramirez, Rafael. 1986. *Action Based Modes of Inquiry and the Host-Researcher Relationship*. 1986.

Golding, David. 2008. *Beginning CakePHP*. s.l. : aPress, 2008.

2011. Internet 2010 in numbers. *Pingdom*. [Online] 1 12, 2011.
<http://royal.pingdom.com/2011/01/12/internet-2010-in-numbers/>.
2012. Job trends. *Indeed*. [Online] 2012.
<http://www.indeed.com/jobtrends?q=zend%2C+cakephp%2C+yii&l=>.
- Joomla. *Joomla*. [Online] <http://www.joomla.org/about-joomla.html>.
- Krasner, Glenn E. and Pope, Stephen T. 1988. *A cookbook for using the Model-View-Controller user interface paradigm in Smalltalk-80*. 1988.
- Lubarsky, Anatoly. 2007. *Facebook Object-Oriented Diagram*. [Online] 2007.
<http://blogs.x2line.com/al/archive/2007/06/02/3124.aspx>.
- Marston, Tony. 2007. The Singleton Design Pattern for PHP. *Tony Marston*. [Online] 3 10, 2007. <http://www.tonymarston.net/php-mysql/singleton.html>.
- Morgan, Terri, Luttrell, Carol and Liu, Yuzeng. 2001. *Designing Multilingual Web Sites: Applied Authoring Techniques*. s.l. : ACM Digital Library, 2001.
- O'Brien, Rory. 1998. *An overview of the methodological approach of action research*. Toronto : s.n., 1998.
2011. October 2011 Web Server Survey. *Netcraft*. [Online] October 6, 2011.
<http://news.netcraft.com/archives/2011/10/06/october-2011-web-server-survey.html#more-5024>.
- Porebski, Bartosz, Przystalski, Karol and Nowak, Leszek. 2011. *Building PHP Application with Symphony, CakePHP and Zend Framework*. s.l. : Wiley Publishing, Inc., 2011.
- Sachdev, Puneet. 2007. Web-Application Internationalization. *Microsoft MSDN*. [Online] 12 2007. <http://msdn.microsoft.com/en-us/library/cc168605.aspx>.
2011. Social media war for video. *Online Marketing Trends*. [Online] 9 7, 2011.
<http://www.onlinemarketing-trends.com/2011/09/social-media-war-for-video.html>.

Sweet, Richard E. 1985. *The Mesa Programming Environment*. California : s.n., 1985.

Synodinos, Dionysios. 2007. Choosing technology/language/framework for web application is not rocket science. *Synodinos*. [Online] 1 3, 2007.

<http://synodinos.net/2007/01/03/choosing-technologylanguageframework-for-web-application-is-not-rocket-science-but/>.

2012. Tiobe programming community index. *Tiobe*. [Online] 2012.

<http://www.tiobe.com/index.php/content/paperinfo/tpci/index.html>.

Welling, Luke and Thomson, Laura. 2009. *PHP and MySQL Web Development*. s.l. : Pearson Education, Inc., 2009.

2009. Which PHP framework holds a promise for the future. *Open source universe*. [Online] 2009. <http://www.opensourceuniverse.com/component/poll/17-which-php-framework-holds-a-promise-for-the-future.html>.

Zend Framework Quickstart. *Zend Framework*. [Online]

<http://framework.zend.com/manual/en/learning.quickstart.intro.html>.

APPENDICES

APPENDIX 1: List of interviewed questions

Decision making:

- Who proposed the idea of adopting web framework into your existing web application?
- What were the reasons of adopting framework to the existing web application?
- What were the factors considered when adopting framework?
- What were the factors considered when choosing that specific framework?

Framework implementation process:

- Who were involved in the process?
- How the implementation was carried out?
- What were the factors influencing the process?

Evaluation:

- How would you evaluate the new application?
- Is it a successful or failure? Why?

APPENDIX 2: Logos of frameworks appeared in alphabetical



APPENDIX 3: Database design of Facebook (Lubarsky, 2007)

