

Advanced Web 2

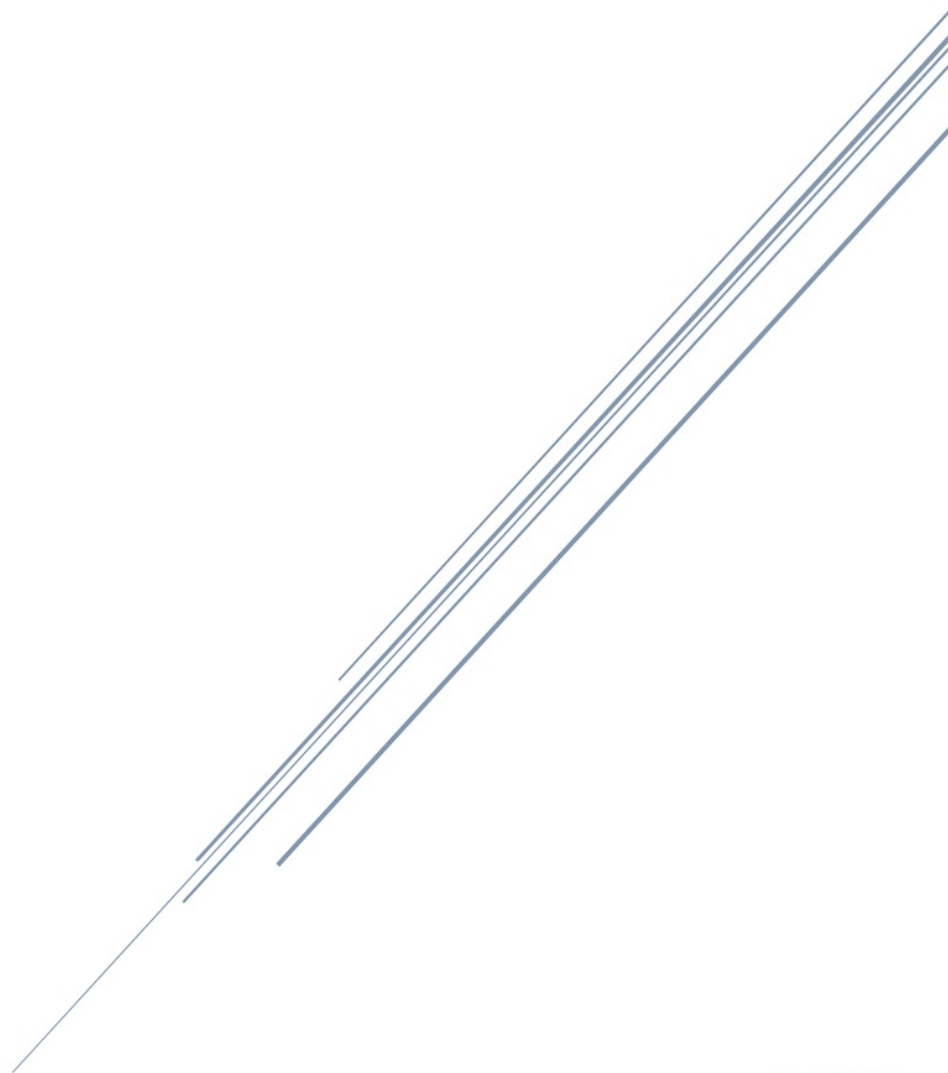
by Thomas Evans

FILE	2770_THOMAS_EVANS_ADVANCED_WEB_2_178145_707344938.DOCX (67.83K)		
TIME SUBMITTED	15-APR-2019 02:40PM (UTC+0100)	WORD COUNT	1373
SUBMISSION ID	104803718	CHARACTER COUNT	7478

ADVANCED WEB DEVELOPMENT

PART 2

Mike Dacey



UWTSD
Web Development

Introduction

This report will be split into two different sections. This first half, knowledge and understanding skills will explore Fowlers' concepts from his book "Pattern of enterprise application Architecture". The report will elaborate on some of his statements and evaluate the use of page front controllers and its use in an MVC framework. The section will also argue the approach of symphony and may offer other alternatives.

The second section of the report will cover the application created to accompany the report. The section will address the concepts covered in the application as well as possible concepts that could be addressed for the project. The section will also cover the software's testing.

Knowledge and understanding Skills

The brief for this report tasks the author of this report with the task of elaborating on a statement from fowlers book titled "Pattern of enterprise application Architecture". The statement reads as followed:

"In a complex Web site there are many similar things you need to do when handling a request. These things include security, internationalization, and providing particular views for certain users. If the input controller behaviour is scattered across multiple objects, much of this behaviour can end up duplicated. Also, it's difficult to change behaviour at runtime"^[1]

This report will cover a series of things mentioned in this quote first of all the security of a complex website compared to that of a request. [2] claim that security is the most significant thing when developing web applications and that now more than ever it is most significant due to a rise in cyber-crime.

Secondly mentioned in the quote is the use of internationalization in web applications. [3] mentions in their paper that maintain a single-point-of-change is crucial when creating large web applications as it can restrict the duplication of code and the use of cut-and-paste to reach a deadline within time. They also go on to mention that the localisation of an application can further lead to the duplication of code segments within an application.

When using the MVC design patterns there is a choice of design patterns that can be used for the controller part of the application. The two different forms of controllers are the page controller and the front controller. The following few paragraphs will explain these patterns in detail and explain the advantages and disadvantages of using either in conjunction with the MVC design pattern.

First the for ease of implementation the PC (Page controller) is more suited for use in simpler web applications due to its low complexity. On the other hand the FC (Front controller) is far more complex and can be harder to implement [5].

An important this to consider when building a large scale web application is code duplication. Code duplication can lead to errors and the less duplication code the better. When using a page controller, the code duplication can be high on the other end of the spectrum the front controller tends to create low levels of code duplication. This can be put down to most common tasks being able to be stored in the front controller [5].

Testability is important for the success of any software and is no different when considering the difference between the different controllers. The page controller has low testability. The only form of testability it has would be with a two-way controller with the http dependent one being separated from an independent controller which is testable. The front controller can only handle requests of independent commands and therefore is testable [5].

Adaptability is important for developing using the MVC pattern. The page controller can contain pages that are very different from each other and therefore can contain high levels of code duplication. This can lead to the application growing too complex

and leave it being less flexible and adaptable. On the other hand, the front controller is highly configurable and therefore used with many CMD frameworks to increase flexibility [5].

For security the page controller the page controller may handle requests for the same page and therefore using static instances may improve the thread safety. The front controller may initiate new command objects for each request to ensure thread safety although database calls should be avoided in the front controller [5].

Through studying both patterns it can be seen that the front controller tends to yield greater results but requires more time and effort to implement correctly. It also requires more skill to initiate. The page controller is the simpler of the two to implement but suffers in most aspects when compared to the front controller.

The brief for this report requires the author of this report to investigate and describe the Template pattern. The template pattern uses an abstract class to outline a set of methods for its subclasses. These methods can then be overwritten in the subclasses for the programmers needs. An example provided in *figure 3 [6]* shows the abstract class Game. The game class contains variables for start play, end play and play. These are then overwritten in the subclasses 'cricket' and 'football' for example play can be overwritten to print "Playing cricket" on the cricket class and "Playing football" on the football class [6]. 

When using these classes, the programmer would be able to create new game of cricket or football but wouldn't be able to create new game. This is due to the original class being abstract.

The brief for this report tasks the author with discussing the approach taken by symphony. Symphony is a PHP web application framework. Symphony as standard does not provide a component to with the ORM database and therefore does not allow access to ORM tools. Although Symphony does allow integration with doctrine which can allow access to ORM tools [7].

When it comes to testing symphony once again integrates with an external framework to allow these capabilities. In this instance symphony integrates with the PHPUnit. PHPUnit is an extensive testing framework used for PHP [8]. Symphony can also utilise a variety of server side templates like Twig. The FOSRestBundle allows for the creation of REST API's using symphony.

Practical Skills

This section of the report will explore a variety of concepts in the form of an MVC application using PHP. Some of the concepts that could be explored include "convention over configuration", MVC architecture or ORM tools among other that could be included.

The application created to accompany this report utilises the MVC design pattern. The application uses a series of controllers, libraries, views and models to create the application.

Sadly, this version of the application does not include any of the suggested concepts from the brief for this report. If the application were to be revisited there would clearly need to be improvements. At the moment the application is simply a working model of an MVC but does not have any other purpose. If the application were to be revisited it should be used to access data from a database as demanded in the brief.

None of the suggested concepts suggested for consideration were used in the application apart from traditional MVC architecture.

If the project were to be redone the concepts that would be included would likely be the integration of a testing framework to help with the applications testing as well as adding in some integration of a REST web service by importing data from an external REST API.

Reference's

- [1] Fowler, M. (2003). *Patterns of enterprise application architecture*. Boston: Addison-Wesley.
- [2] Aihua Jendy Wu. (2014). Project Development for Ethical Hacking Practice in A Website Security Course. May 2014.
- [3] Terence Parr(2006) Web Application Internationalization. July 2006.
- [4] Luo GuangChun, WangYanhua Lu, Xianliang Hanhong. (2003). A novel web application frame developed by MVC. March 2003.
- [5] Pilaka, P. and profile, V. (2019). *Page Controller Vs Front Controller Pattern*. [online] Pratapreddypilaka.blogspot.com. Available at: <http://pratapreddypilaka.blogspot.com/2011/09/page-controller-vs-front-controller.html> [Accessed 3 Apr. 2019].
- [6] www.tutorialspoint.com. (2019). *Design Patterns Template Pattern*. [online] Available at: https://www.tutorialspoint.com/design_pattern/template_pattern.htm [Accessed 6 Apr. 2019].
- [7] Symfony.com. (2019). *Databases and the Doctrine ORM (Symfony Docs)*. [online] Available at: <https://symfony.com/doc/current/doctrine.html> [Accessed 7 Apr. 2019].
- [8] Symfony.com. (2019). *Databases and the Doctrine ORM (Symfony Docs)*. [online] Available at: <https://symfony.com/doc/current/doctrine.html> [Accessed 7 Apr. 2019].

Table of figures

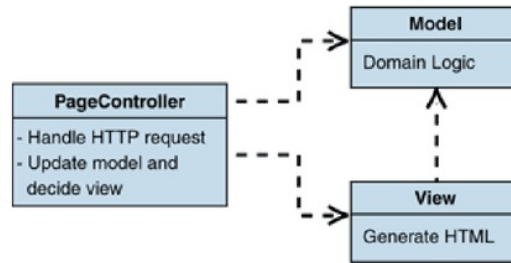


Figure 1: Page controller

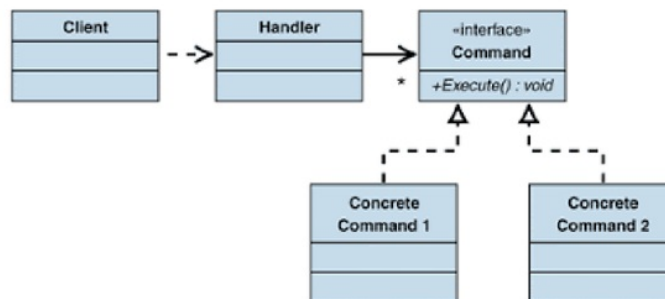


Figure 2: Front Controller

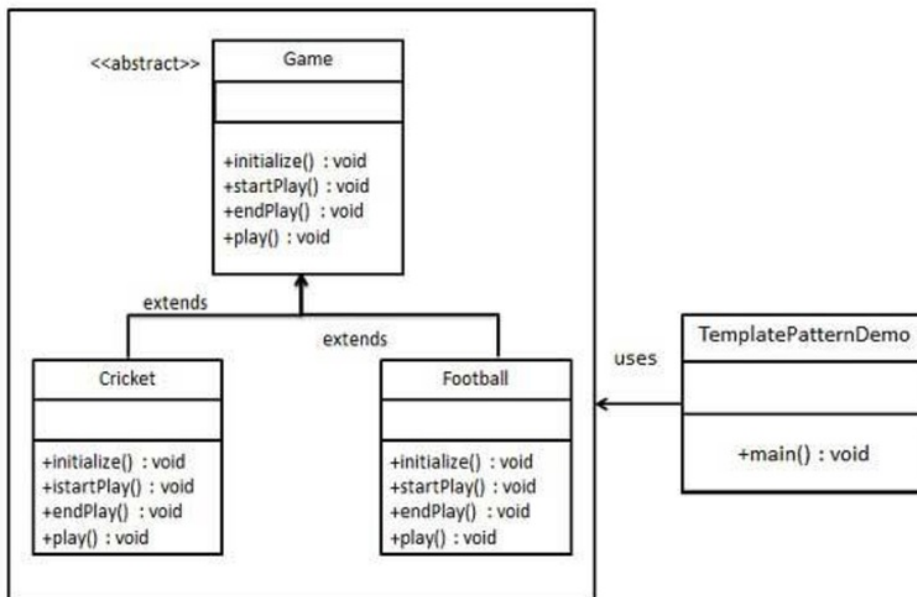


Figure 3: Template pattern [6]

FINAL GRADE

35/100

GENERAL COMMENTS

Instructor

Investigation of Fowler (30) 17

Investigation of Symfony (20) 9 Quality (20) 6

View handling (10) 3

Database handling (15) 0

Software testing (5) 0

LATE CAPPED 40

Good evaluation of Fowler. You did didn't evaluate the TemplateView but I will give you marks for the pattern you did describe.

The discussion of Symfony is very brief and with little evaluation.

You have attempted to create your own MVC like framework rather than use the one provided in class. I'm not sure why. Unfortunately, there are many errors in your version, some of them quite basic such as spelling controller with 3 Ls. I will give you some marks because some of the code is interesting.

PAGE 1

PAGE 2

PAGE 3

PAGE 4



Comment 1

Not quite, in the context of enterprise patterns the Template View was the pattern that I intended you to discuss. I will give you some marks, as the pattern you describe could be applicable.

PAGE 5

PAGE 6

PAGE 7
