

**ASSIGNMENT 1**

**ISS2263 – SOFTWARE DESIGN AND ARCHITECTURE**

*MADAM AZALIZA BINTI ZAINAL*

AFIFFUDDIN NOLI (4182012231)

NASRULLAH BIN ZAINI BAKRI (4182005921)

**What is the differences between PHP programming using MVC design pattern and without using MVC design pattern. Support your answer with a literature review**.

When we create a software, one of the main problem is to organize the source code. We want it to be:

* Easily maintainable
* Reusable
* Understandable
* Easily to work on it with many different people on the same time.
* Many different generations of developers had this problem and reflects on a solution to regulate this. When one of these strategy can be reusable and resolve some recurrent architecture problems with predefine rules, we can name it a design pattern.
* A design pattern is a description of standardized solution to resolve some architecture problem when designing a software. It does not depend of languages or environments.
* Everyone can define his own design pattern, but some researcher and developers with huge experience reflects on the problems and found some solutions used by many developers because they are reliable and practical.

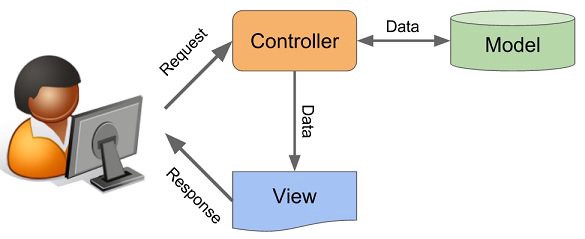
The MVC design pattern.

MVC is an abbreviation for 'Model View Controller' and it is one of the most known and used design patterns. With an MVC architecture, we split the code on three distinct parts:

- The Model part will manage the data access and their processes.

- The View part will manage the display at the HMI (Human Machine Interface) level

- The Controller part will be the event handler. He will receive all action executed by the user and will execute the needed operations by calling the appropriates views and models. It is the link between the views and the models. The controller is the central piece of the MVC architecture.



This split of the code has many advantages:

- Each parts can be created or edited by different people with different expertise.

- It is easier to maintain and edit the project. We can edit one specify part without affect other part and we can know quickly where are the different elements.

Using the MVC design pattern.

For each project view, we have to define:

- The dynamic parts.

- The data needed by the controller.

With that we know the processes that we have to create on the model.

-> The data will be send to the controller by the view via the forms (GET or POST methods).

-> The model will implement all the processes that can be launched from the view, as well as all the processes that allow to access at the data needed to display the view.

-> The dynamic data will be send by the controller to the view:

- Controller can do an include of the model, so it will can execute the processes created on the model

- The controller can put the process result on predefine variables.

- The controller can do an include of the view.

The view can access the predefine variables needed for the display. The view can be represented as a page with inputs (data provided by the controller) and outputs (the data sent to the controller via the forms).

**WITH MVC PATTERN**

1. **Faster development process:**

MVC supports rapid and parallel development. If an MVC model is used to develop any particular web application then it is possible that one programmer can work on the view while the another can work on the controller to create the business logic of the web application. Hence this way, the application developed using the MVC model can be completed three times faster than applications that are developed using other development patterns.

2. **Ability to provide multiple views:**

In the MVC Model, you can create multiple views for a model. Today, there is an increasing demand for new ways to access your application and for that MVC development is certainly a great solution. Moreover, in this method, Code duplication is very limited because it separates data and business logic from the display.

3. **Support for asynchronous technique:**

The MVC architecture can also integrate with the JavaScript Framework. This means that MVC applications can be made to work even with PDF files, site-specific browsers, and also with desktop widgets. MVC also supports an asynchronous technique, which helps developers to develop an application that loads very fast.

4. **The modification does not affect the entire model:**

For any web application, the user interface tends to change more frequently than even the business rules of the .net development company. It is obvious that you make frequent changes in your web application like changing colors, fonts, screen layouts, and adding new device support for mobile phones or tablets. Moreover, Add a new type of view are very easy in the MVC pattern because the Model part does not depend on the views part. Therefore, any changes in the Model will not affect the entire architecture.

5. **MVC model returns the data without formatting:**

MVC pattern returns data without applying any formatting. Hence, the same components can be used and called for use with any interface. For example, any kind of data can be formatted with HTML, but it could also be formatted with Macromedia Flash or Dream viewer.

6. **SEO friendly Development platform:**

MVC platform supports the development of SEO friendly web pages or web applications. Using this platform, it is very easy to develop SEO-friendly URLs to generate more visits from a specific application. This development architecture is commonly used in the Test-Driven Development applications. Moreover, Scripting languages like JavaScript and jQuery can be integrated with MVC to develop feature-rich web applications.

Thus, the MVC design pattern is surely a great approach to building software applications. The MVC framework is easy to implement as it offers above given numerous advantages. Projects that are developed with the help of the MVC model can be easily developed with lesser expenditure and within less time too. Above all, its power to manage multiple views makes MVC the best architecture pattern for developing web applications.

As a result, today organizations are looking for the .net development of web applications based on MVC architecture for cost and time benefits. There are many web development companies providing MVC development services to develop web applications that satisfy every requirement of the clients. Brainvire is one such dot net development company that provides the most desired output to its clients by offering fast and highly interactive web applications using MVC 6 development architecture.

**WITHOUT USING MVC PATTERN.**

The pattern behind every screen we use is MVC –Model-View-Controller. MVC was invented when there was no Web and software architectures were, at best, thick clients talking directly to a single database on primitive networks. And yet, decades later, MVC is still used, unabated, for building OmniChannel applications.

If want to use MVC without a framework, all need to do is divide the PHP files by their functions (controllers, models and views,) store them in separate directories and make sure they know how to find each other. But then will have written a simple framework

**Comparison Between Core PHP & MVC Framework**

Development Quality: PHP frameworks like Laravel & Symfony are the most popular framework in the market these days, in terms of development quality Frameworks can overcome the core PHP source code. We have some of the application where the developer writes the complete logic for multiple functionalities in just one file the length of that file goes in thousands, that code consists of HTML, PHP, JavaScript and Style all together in just on a single file. Which makes it difficult to understand the code if anything goes wrong in the functional flow. On the other hand, Using an MVC framework segregates the code into different layers like Model, View and Controllers.

Easy Modification, Manipulation & Customization: It’s very easy to find out the exact place for the logical error if we are going to use an MVC framework because we can get to know the controller and the model via its route and debug at the right place. Latest frameworks like Laravel and Symfony provides us with the inline command tools which make our work even easier. But if we have used Core PHP and haven’t managed the files properly, in a separate folder structure with a proper hierarchy, then the developer will be in trouble, either it is a small or time taking modification, developer will have to spend the maximum time unnecessarily, to find out the place because for Core Language there is no predefined architecture, but for frameworks, it is there. You can search the documentation for every framework available, and if the development has been done according to the development guide it will be a child’s play for a developer to make the required changes.

Database Manipulation & Migration: Database Manipulation can be like; creating a new table, removing an existing table, field alteration, etc. Which should be done parallelly in the respective code base, otherwise the application will stop working. So, if we are working with core PHP we need database access along with that we will have to do the changes manually in our database structure by login into the cPanel etc. And it would be difficult if we need to migrate a complete database structure from one server to another, for doing this we will have to export the SQL file from server and import into another server, for this kind of work we would need a database administrator. But if we have developed our application using an MVC framework like Laravel or Symfony, Database Manipulation will be a very easy task, because these kinds of frameworks provide us with the feature for database migration, the project consists with a folder where all the migration scripts reside. And using few commands on terminal we can run those migration scripts which can create and alter the table structure, dummy data can also be seeded into multiple tables in just one go.

Secure Introspectable Tunnels: A developer always needs and online space to display their work to their client, remote team members etc. the same situation happens when a developer is using core PHP instead of using an MVC Framework, He needs an online space to upload his project files then he needs to configure the database if the application is data-driven. Bearing a server cost during the project development is like an investment in the project, but what if we can show the project to the client without pushing it to any server, this will be a great thing to know. Yes, Our PHP Framework like Laravel provides us with a secure Introspectable Tunnel for localhost, which makes it possible to provide an online URL of your project to anyone, anywhere in the world. But if you are still thinking to develop your application using Core Language then you will have to bear the server cost during the development of your project.

ORM Vs Raw SQLs: A big ruling thing under the Frameworks Architectures is the Integrated ORMs, like Eloquent etc. which provide the complete set of rules and functionality to play with intense SQL operations, CRUD, Transections, etc. Eloquent provides a systematic way to perform the data-driven task in the project, on the other hand, it helps to be secured from the SQL Injections as is sanitize all type of request what an MVC Framework receives from the internal system or the external one. With Core PHP we use to write all the raw SQL queries for database operations, which waist a lot of time to prepare the correct & optimized query.

Conclusion

It may seem that the task of researching the use of design patterns in the design of Web applications is simple, but as soon as the work begins, it becomes clear that the situation is far more complicated than it has been expected. Some frameworks have good homepages, which most framework developers have not really ensured. Hence, finding information about projects can be very complicated or even impossible. Most developers have taken the time to prepare sensible comments in relation to their software codes, but sometimes it is quite hard to understand whether the method has a name because a specific design pattern has been used, or the developer has simply decided on what he or she considers to be the most appropriate name without even thinking about the design pattern. After collecting all the information, we found that MVC and Singleton are still the leading design patterns. That is logical because the separation of data, business logic, and visualisation are the basic ideas in designing complicated systems. Other design patterns which are frequently used include Application Controller, Builder, Front Controller, Iterator, Observer, Registry, and State. However, these are not the only design patterns used, which reminds us once again how very diverse the frameworks are – each uses a different set of design patterns which suggests that developers have encountered various problems during the development process. Further research could focus on drafting the design for a framework which uses as many design patterns as possible, taking into account the experience in the use of design patterns in the surveyed projects. This could show the strength of design patterns in the way that most of the functionality could be described through known design patterns while new design patterns could be developed for the rest. Alternatively, researchers might find out that not everything can be described with the help of design patterns.

Still other alternative might be the attempt to create and study a completely functional open code project such as the E-store, which is based on a certain framework. The analysis of such a project might allow researchers to identify the real problems that can be resolved or hindered by the use of design patterns. When we analysed the use of design patterns, we found it necessary to group all design patterns into logical groups because it is hard to manage a large number of design patterns which each is completely independent; therefore, it is also necessary to indicate the links and interaction among various design patterns. In conclusion, we must also say that the success of a project does not depend on the use or omission of a design pattern. There are frameworks which use very few design patterns while others use many of them. It can never be claimed that this fact accounts for one framework being better or worse than other.

**References :**

<https://www.supinfo.com/articles/single/6019-php-and-mvc-design-pattern>

<https://www.sitepoint.com/the-mvc-pattern-and-php-1/>