

Applying an MVC Framework for The System Development Life Cycle with Waterfall Model Extended

W Hardyanto^{1,*}, A Purwinarko², F Sujito², Masturi¹, and D Alighiri³

¹ Pysics Department, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

² Computer Science Department, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

³ Pysics Department, Faculty of Mathematics and Natural Sciences, Universitas Negeri Semarang, Indonesia

*Corresponding author: hardy@mail.unnes.ac.id

Abstract. This paper describes the extension of the waterfall model using MVC architectural pattern for software development. The waterfall model is the based model of the most widely used in software development, yet there are still many problems in it. The general issue usually happens on data changes that cause the delays on the process itself. On the other hand, the security factor on the software as well as one of the major problems. This study uses PHP programming language for implementation. Although this model can be implemented in several programming languages with the same concept. This study is based on MVC architecture so that it can improve the performance of both software development and maintenance, especially concerning security, validation, database access, and routing.

1. Introduction

Web application development is always growing rapidly through the years. It requires the system to work more quickly and accurately. Many developers are trying to create new and different technologies, especially for building a faster and more powerful web applications. Until now, web application like front-end or client-side still using HTML, CSS, and JavaScript to present data. While the back-end or server-side refers to the storage and processing of data that normally handled by PHP.

Today, information technology is one of the fastest growing businesses areas. The software contains three major modules, namely the interface, business logic, and data. All three are closely related so that software developers have to write all the code for production applications. Currently, with the MVC design, developers do not have to do that because it serves to separate the input, business logic, and output and it also makes model classes reusable without modification.

SDLC model consists of a sequence of steps that must be followed by system developers. Therefore, many companies engaged in software development chose this model to cause it is easy to make applications with high quality. One of the SDLC models, namely the Waterfall, which consists of five subsequent stages, there are business analysis, design, implementation, testing, and maintenance. The success of the Waterfall model makes many software development companies adopting the main framework for the design, build, and maintain an application.

This paper proposes the concept of MVC for software development using SDLC techniques, especially the expansion of the Waterfall model. The goal is to optimize the Waterfall models with



MVC architecture, so it will get higher quality applications regarding safety and minimizes processing time.

2. Methods

2.1. Waterfall Method.

In 1970, one of the articles Walter Royce [1] is the first publication that discusses the waterfall model. Older models widely criticized and modified so as to produce new models and different. The waterfall model of designing things related to software development before starting coding and to successfully build the system. The main function of the waterfall model is to ensure the success of the project. Waterfall method consists of several phases of the software development process, and there are requirement analysis, design, implementation, testing, and maintenance.



Figure 1. The stages of Waterfall Model

2.1.1. Requirements Analysis

This stage called Software Required Specification (SRS). The main focus at this stage is a complete description of the software to be developed. The project manager should discuss with users of the system to analyze the functional and nonfunctional requirements for the fulfillment of this stage. Functional requirements typically include the purpose, scope, perspective, function, user characteristics, interface, and database. Whereas the nonfunctional requirement is referring to criteria, constraint, limitation, and software performance[2].

2.1.2. Design

At this stage not only about interface design, but also algorithm design, software architectural design, database, concept design, and structural design. System and software design are made of the requirement specification that is learned in the first phase. The system design help in determining hardware and defining the overall system of architecture.

2.1.3. Implementation

This stage is the realization of business requirements and system design into the application program. Database starts with coding and deploys it. At this stage, the real code was written and compiled into an operational application, where the database and other needs have been learned in the previous stage. The work is divided in modular form to facilitate the development as teamwork. The focus at this stage is the development of software so that this stage is the longest stage of the SDLC[11].

2.1.4. Testing

It is also known as the verification and validation that is a process to check that the software meets the original requirements and complete the objectives that have been set. In fact, verification is a software evaluation process to determine whether a product of a given development phase satisfy the conditions imposed at the beginning of that stage. Meanwhile, validation is the process of evaluating software during or at the end of the development process to determine whether it fulfill the requirements specified or not[3]. Also, the testing phase is an outlet for debugging where bugs and system disorders discovered, corrected and improved.

2.1.5. Maintenance

It is the process of modifying the software after it was created and deployed, correcting errors, and improve performance and quality. Additional maintenance activities can be done in this phase includes software adaptation to its environment, to accommodate the needs of new users, and improve software reliability.

2.2. MVC Framework.

MVC design pattern was first proposed by Trygve Reenskaug in 1970 at Xerox Parc. According to him, "an important goal of MVC is to bridge the gap between the user model and the digital model on the computer[4]. Then, in 1988, the MVC paradigm described in detail by Krasner and Paul in their articles "A cookbook for using the model-view controller user interface paradigm in Smalltalk-80", which was published in the Journal of Object Oriented Programming.

They emphasized there be great benefits that can be obtained when people build applications with modularity. "Isolating functional units from one another as much as possible makes it easier for application designers to understand and modify any unit without having to know everything about the other units"[5]. MVC pattern is divided into three main roles allowing for more efficient collaboration. The main role here is the development, design, and integration.

The role of development taken by experienced programmers who are responsible for application logic. They take care of the data query, validation, processing, and others. The role of designers to developers who are responsible for the look and user-friendly applications. They display data supplied from the developers who worked on the first role. MVC design pattern is very suitable for the development of web applications because they combine some of the technology that's usually divided into a set of layers[6]. Figure 2 shows the relations between the components of the MVC architecture.

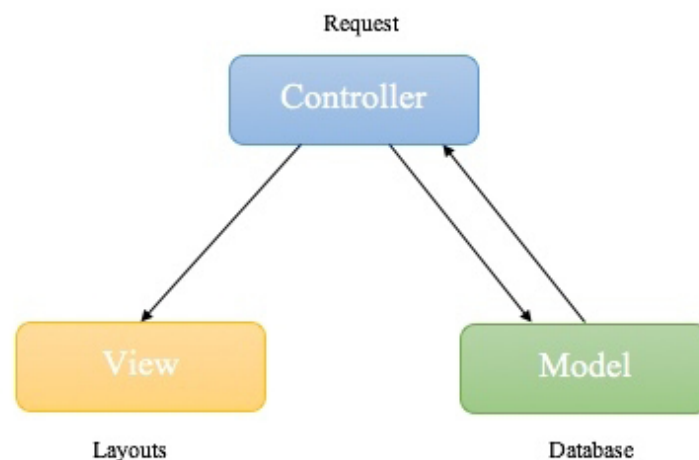


Figure 2.MVC Pattern [4]

2.2.1. Model

Model is part of a system that manages all things that associated with the data, validation, control, and structure of the data source (database). The model greatly reduces the complexity of code developers in writing[7]. Model responsible with the application business logic. This will serve to access the data (database, file, etc.) and will create a class that can be reused. The model usually builds with the data abstraction in mind, validation, and authentication[8].

2.2.2. View

The view is responsible for the management graphical user interface. This means all forms, buttons, graphic elements and all other HTML elements that are within the application. Views can also be used to generate the RSS content to the Flash presentation. By separating the application design from application logic, it greatly reduces the risk of human error when the designer decides to change the interface of the application. At the same time, job application developers greatly reduced because they no longer need to see the HTML code elements, design elements and graphic elements[7]. A view is usually called a web design or template. It controls the way data is displayed and how the users interact with it. It also provides a way to collect data from users. The technology that mainly used is HTML, CSS, and JavaScript. As a general rule, a view should not contain elements that include application logic, to facilitate designers to work with it. The logic blocks should be kept to a minimum.

2.2.3. Controller

The controller is responsible for handling the event. This can be done by one of the users who interact with the application. A controller receives the request and prepares the data to respond. It is also responsible for establishing the response format. The controller interacts with the model to retrieve the necessary data and produces View. This process is also known as action or verb[7]. When a request arrives at the server, the MVC framework sends to a method on a controller based on the URL[9]. The controller has all of the application logic and incorporates display in View with functions in the model. It is responsible for the retrieval of data from the View and to build a path for application execution. The controller will access the function model and interprets the received data that can be displayed by the View. It is also responsible for error handling[8]. A Controller manages the relation between the View and the Model. It responds to user requests, interact with the Model and View to decide what should be created and displayed[10]. As what mentioned above, we need an approach to Controller where all data processing applications should be dealt with at the level of the Controller. Our control systems can support JSON, text and XML data format for requests and responses.

2.3. Design and Implementation.

In this section, it describes the architecture of the design and operation of the framework. This is a three-level application framework that arranged into three major sections.

1. Users
2. MVC Components
3. Database

The first level consists only of users. Users can send a request and a response by components of the "View". The second level consists of three components called Model, View, and Controller. These components can communicate and process data with each other. Finally, the third level only database where data is stored permanently. Figure 3 illustrates how data flows throughout the system using MVC.

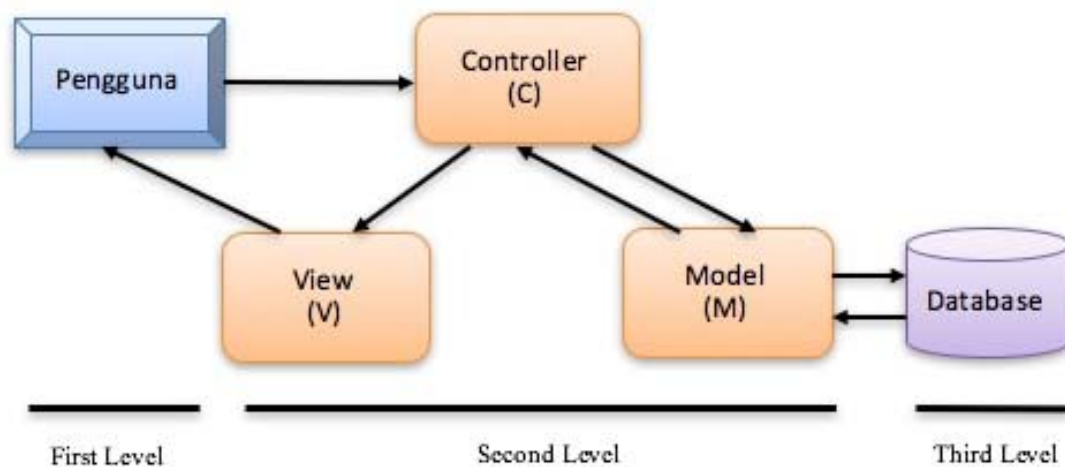


Figure 3.MVC Arsitektur [4]

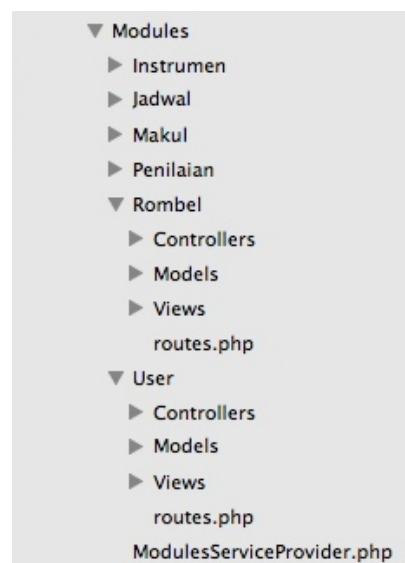


Figure 4.Folder Organization

Here is the explanation of the operational of the application.

1. The user provides a request via typed URL. Routing in the system works to continue to the Controller. Controller processes user requests. Based on user demand, the Controller will call the View and the Model. The Model is used if the user requests regarding data stored in a database. The Controller adjusts the demand for the model. The model represents the entire data and application logic.
2. Then the Controller continues it to the View. The view is responsible for the output of the model. The graphical display of pure contained in this process. View displays a request given by the user. So the users get the results they want.

2.4. Waterfall Model Extended

The main focus of this research is on the expansion of the waterfall model. The waterfall phase, in general, is divided into five stages. But the concept that we propose is the expansion of the waterfall

model remain in 5 stages but has a different groove. Also, there is a stage in its development using MVC architecture which is implementation stage. Gambar stage 5 shows how the lane of the waterfall model with the expansion of the use of MVC architecture.

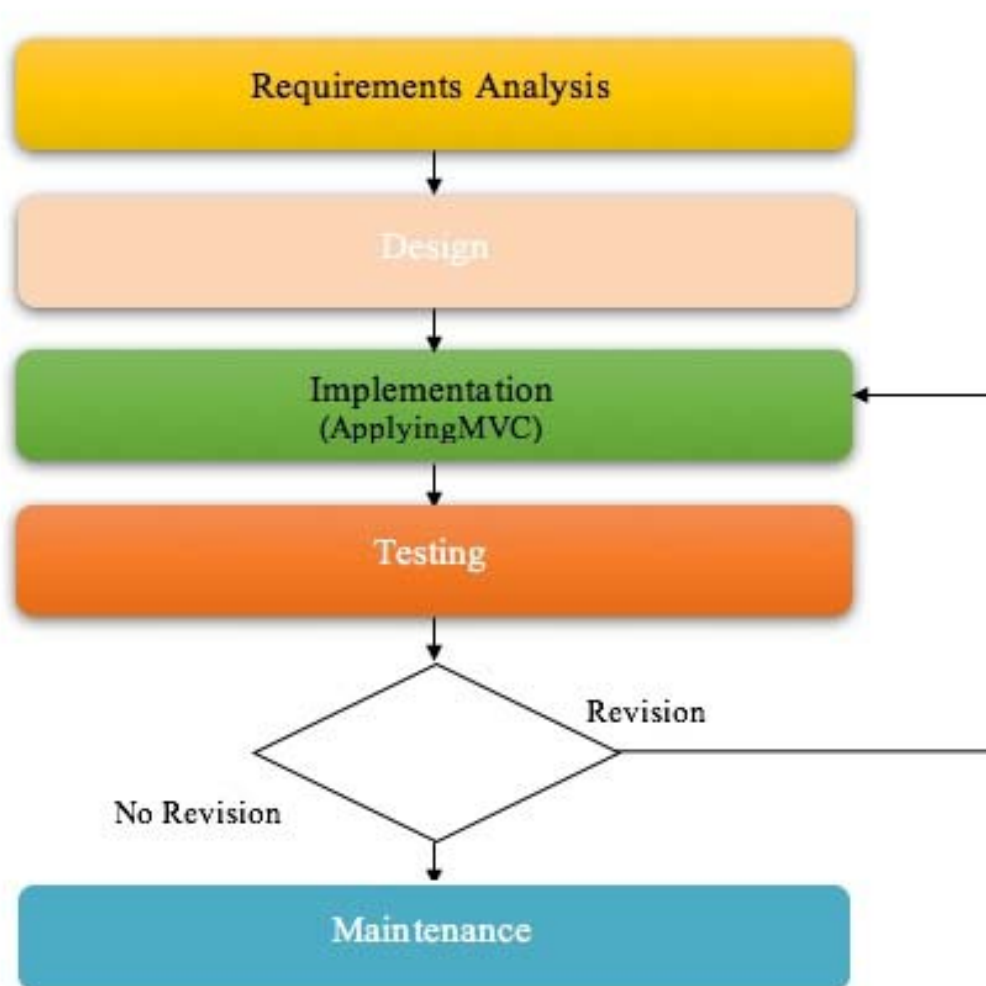


Figure 5. Expansion of the waterfall model and the application of MVC architecture

The expansion lane of the waterfall model starts from the analysis of needs. After analysis of needs is assessed, then the data will be designed from the user interface, application logic, database, and so on. Later in the implementation phase with the software is written through coding. In its development, we apply the MVC architecture at this stage. Because at this stage of development, there definitely be many obstacles around the business logic, interface, and database. So with the MVC architecture, the problem will be solved. After the application is completed, then we can do a test. If the applications were built without any flaws, then the system can be directly to the maintenance phase without revised. However, if the system is still uncompleted, then the system will return to the stage of implementation and do the revised. Once the system is back to be tested to maximize the application functionality.

3. Results and discussion

Implementation stage is the stage of making the information system according to the results at the design stage. We build a simple application and its called "Microteaching" to test this model.

Implementation of this system is using the frameworkLaravel. Framework laravel facilitate the implementation of the design stage into the system because it uses MVC architecture. The ease of using frameworkLaravel is the separation of program code based on the Model-View-Controller pattern, implementing the functional requirements into separate modules, and the availability of build-in library so as to accelerate the implementation process.

Here, we will show you some pictures of the application. We perform different types of operations, such as entering data into a database, retrieving data, change or update the data, searching data and so on. We get the appropriate results for each step.

Figure 6. Insert Data

Figure 7. Update Data

No	NIP	Nama Dosen	Username	Aksi
1	198509102015041001	AJI PURWINARKO S.Si., M.Cs.	dosen1	[Edit] [View] [Delete]

Menampilkan 1 sampai 1 dari 1 data (difilter dari 2 total data)

Figure 8.Result Of Data

No	NIP	Nama Dosen	Username	Aksi
1	198509102015041001	AJI PURWINARKO S.Si., M.Cs.	dosen1	[Edit] [View] [Delete]
2	198603162012122001	INDAH URWATIN WUSQO S.Pd., M.Pd.	dosen2	[Edit] [View] [Delete]

Menampilkan 1 sampai 2 dari 2 data

Figure 9.Data Search

4. Conclusions

Based on the proposals above, it says that the expansion of the waterfall model is used by many large companies for their internal projects. Due in large companies most clients can change their requirements at any specified time. Not only time, usually the client wants to change the graphical display totally at any time. So it does not waste time and would hold up the process of software

development. However, with the application of MVC architecture in software development, application developers can easily change the parts that need to be changed suddenly. Because this framework separates View, Controllers, and Model. Therefore, this framework is very stable, efficient and able to create high-quality applications.

References

- [1] Patel, U. A., 2013. New Idea In Waterfall Model For Real Time Software Development. *International Journal of Engineering Research & Technology (IJERT)*, April, 2(4), pp. 114-119.
- [2] Bassil, Y., 2012. A Simulation Model for the Waterfall Software Development Life Cycle. *International Journal of Engineering and Technology*, May, 2(5), pp. 742-749. Patel, U., 2013. New Idea In Waterfall Model For Real Time Software Development. *International Journal of Engineering Research & Technology (IJERT)*, April, 2(4), pp. 114-119.
- [3] S. D. Mone, "Comparative study of Waterfall model with RAD model," *International Journal of Modern Trends in Engineering*, pp. 481-484, 2015.
- [4] Trygve Reenska, <http://heim.ifi.uio.no/~trygver/themes/mvc/mvc-index.html>
- [5] Glenn E. Krasner, Stephen T. Pope, "A cookbook for using the model-view controller user interface paradigm in Smalltalk-80", *Journal of Object-Oriented Programming*, vol. 1, no. 3, 1988, pp. 26-49.
- [6] Badurowicz, M. "MVC Architectural Pattern In Mobile Web-Applications", *Actual Problems Of Economics*, 2011, pp. 305-309.
- [7] W. J. Gilmore, *Easy PHP Websites*, Columbus, Ohio: W.J. Gilmore, LLC, 2009.
- [8] A Freeman, S Sanderson, *Pro ASP.NET MVC 3 Framework*, Apress, 2011.
- [9] Stratmann, E., & Ousterhout, J., "Integrating Long Polling with an MVC Web Framework", 2nd USENIX Conference on Web Application Development, 2011, pp. 113-124.
- [10] J. Galloway, P. Haack, B. Wilson și K. S. Allen, *Professional ASP.NET MVC 3*, John Wiley & Sons, Inc., 2011.
- [11] Purwinarko A and Y. L. Sukestiyarno, "Model Expertise Management System di Universitas Negeri Semarang," *Sci. J. Informatics*, vol. 1, no. 2, pp. 177-184, 2014.