



RESEARCH ARTICLE

A Comparative Study of Web Development Technologies Using Open Source and Proprietary Software

David A. Botwe¹, Joseph G. Davis²

¹Department of Informatics, Regent University College of Science & Technology, Ghana

²Department of Computer Science, Kwame Nkrumah University of Science & Technology, Ghana

¹ davebotwe@yahoo.com ; ² omanyi054@yahoo.com

Abstract— A web application is a distributed application that runs on more than one computer and communicates through a network or a server. Specifically, a web application is accessed with a web browser as a client and provides the ability to update and maintain a program without deploying and installing software on client computers. Web programmers today are confronted with the difficulty of having to work with constantly changing technologies, and having to make the right choice of which development technology to use. This paper presents a comparison of web application development technologies using open source software and proprietary software. The comparison involves three major web development technologies namely: Java Server Pages (JSP), Active Server Pages (ASP.NET) and PHP Hypertext Preprocessor (PHP). For the comparison to take place, a web application was developed in all three technologies using the same requirements. The model-view-controller (MVC) design pattern was employed in developing the web application, and the parameters for the comparison are cost of implementation, browser compatibility, response time of HTTP requests, operating system compatibility and mobile platform compatibility.

Keywords: Web Application, Open Source, Proprietary, JSP, PHP, ASP.NET, MVC

I. INTRODUCTION

A web application is a distributed application that runs on more than one computer and communicates through a network or a server. Specifically, a web application is accessed with a web browser as a client and provides the ability to update and maintain a program without deploying and installing software on client computers. Web applications are used for web mail, online retail sales, discussion boards, weblogs, online banking, among others. One advantage of a web application is that it can be accessed and used by millions of people at the same time [1].

Web applications usually run on the Internet or an Intranet and they have become an important part of the business world in recent times. With the increasing number of users of the Internet and World Wide Web, more sophisticated Internet and web applications have emerged. The functionality of Internet browsers have been extended to include dynamic elements, such as event handling, processing of forms, performing of calculations, among others. Scripting languages, such as VBScript or JavaScript have been introduced to enhance the mostly static contents of web pages with dynamic elements.

Dynamic HTML (DHTML), extensible HTML (XHTML) and XML are some of the technologies that are part of today's browsers to provide interactivity to web applications. Web servers have also been enhanced to respond to client's requests in a more flexible way than presenting the same content to all users. Server-side scripting is the key technology that allows developers to make Web applications fit for a specific purpose, so incoming requests can be processed by the server on the fly.

With server-side scripting, web pages are prepared by the application (on the server) considering the client's request. The information submitted to the client can be in HTML format, which is the Internet's basic standardized language of communication. Several server-side technologies exist today. Examples are Active Server Pages (ASP or ASP.NET), Java Servlets and Java Server Pages (JSP), PHP (Hypertext Preprocessor), Perl and Python.

The idea of comparing various web application development tools and technologies has been an important consideration for many web application developers. While some researchers use these comparisons to decide which technologies and tools to adopt in developing new web applications others use them for developing new frameworks to simplify the web application development process [2], [3].

This paper presents a comparison of web application development technologies using open source software and proprietary software. The comparison involves three major web development technologies namely: Java Server Pages (JSP), Active Server Pages (ASP.NET) and PHP Hypertext Preprocessor (PHP).

II. STATEMENT OF THE PROBLEM

The major challenge facing Web programmers today is choosing the right language for the job. Not only the language must fit the job but it also must fit the programmer. The most common deciding factor typically centres on the developer's familiarity, which according to [3] is an extremely subjective measurement. Other factors include the cost of implementation, browser compatibility, response time of HTTP requests, operating system compatibility and mobile platform compatibility.

Since the server-side application development is still evolving, developers are confronted with the difficulty of having to work with constantly changing technologies, and having to make the right choice of which development technology to use. While some of the web development technologies are compatible with different operating systems and browsers, others are not.

III. AIM & OBJECTIVES

This paper compares web application development technologies using open source software and proprietary software.

The following are specific objectives that have been set to achieve the above aim.

- To compare the major technological differences and similarities in web development technologies using open source tools such as JSP and PHP and proprietary tools such as ASP.NET.
- To explore the merits and demerits of using open source software and proprietary software in web application development.
- To compare the performance of the various web development technologies on different platforms including mobile devices.

IV. METHODOLOGY

2.1 System Architecture using Java

Figure 1 shows the Model-View-Controller (MVC) design pattern as described by [4]. The key idea was to separate the application into three main components: the model, the view and the controller. The model contains the business logic that processes the application's data. It also stores or retrieves data to or from the database. The view displays information to the user and the controller handles user interactions and input. The views were developed using JSP web pages (*.jsp), the controllers were developed using Java servlets (*.java) and the models were developed using Enterprise Java Beans (EJB) and the Java Persistence API (JPA).

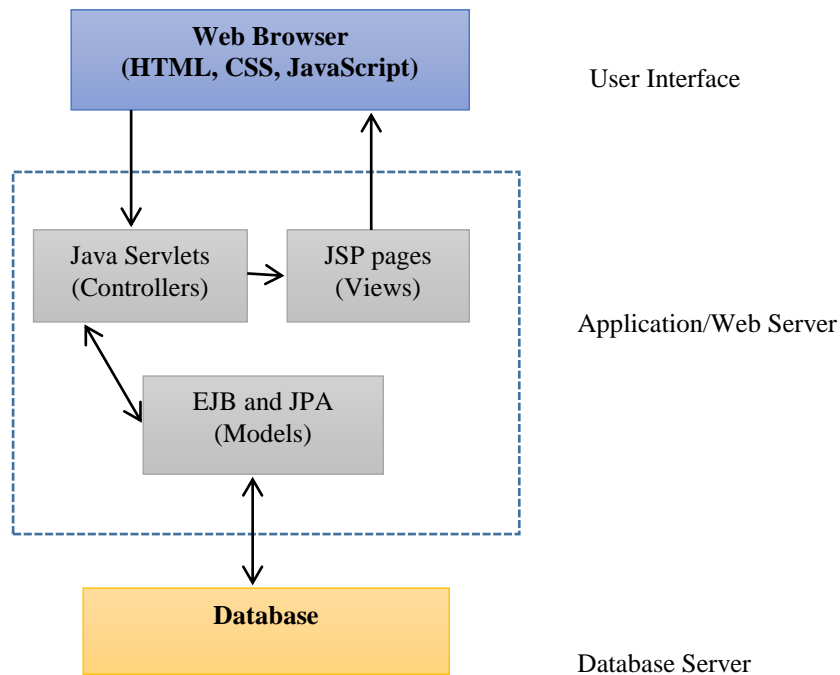


Figure 1: Model-View- Controller Design Pattern using Java

2.2 System Architecture using ASP.NET

Figure 2 shows the Model-View-Controller (MVC) design pattern which was used to develop the application in ASP.NET. This is similar to the MVC design pattern for Java with the main difference being that the view was developed using an Active Server Pages (ASP) and the controller is in C# code. The views were developed using ASP.NET web pages. There are two engines for developing the views, namely razor (*.cshtml) and aspx (*.aspx). The razor engine was used to develop all the views, the controllers were developed using C# classes (*.cs) and the models were developed using Entity Classes which are also C# classes (*.cs).

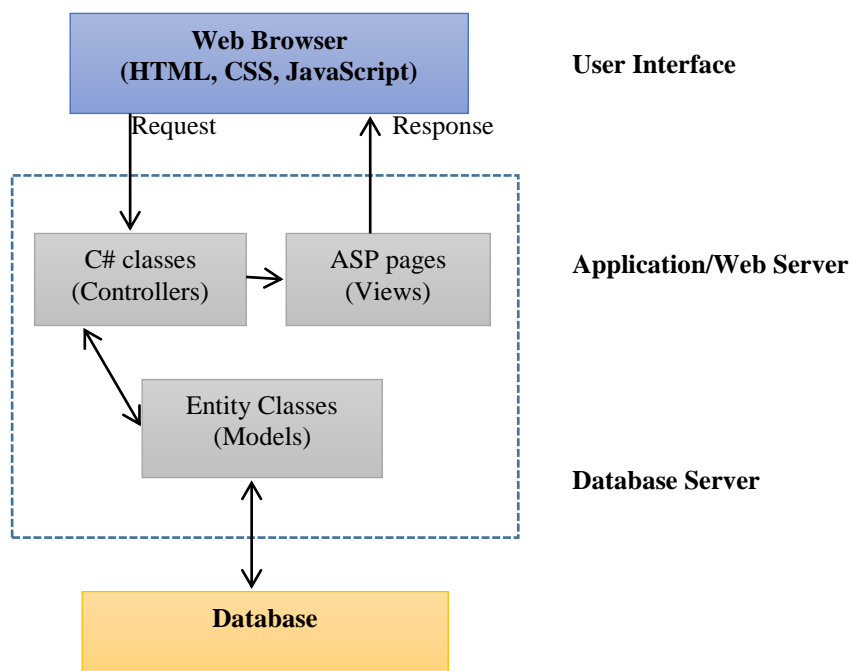


Figure 2: Model-View- Controller Design Pattern using ASP.NET

2.3 System Architecture using PHP

Figure 3 shows the MVC design pattern which was used to develop the system in PHP. The views were developed using PHP web pages and the controllers were developed using PHP classes (.php). The models were developed using Entity Classes which are also PHP classes (.php).

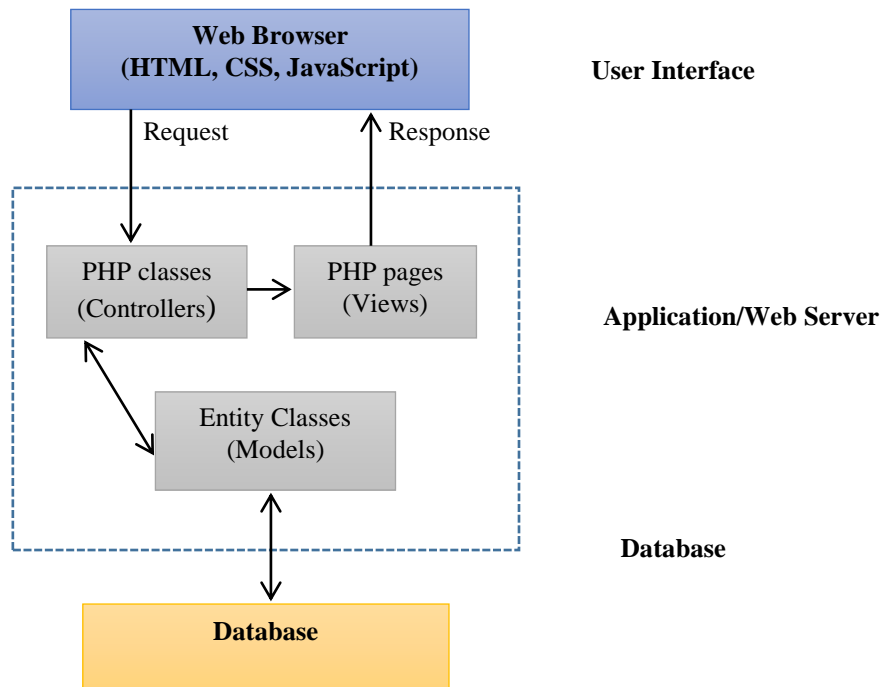


Figure 3: Model-View- Controller Design Pattern using PHP

2.4 Implementation in Java

The system was implemented in Java using Java Server Pages (JSP) for the views, Java servlets for the controllers and Enterprise Java Beans (EJB) for the models. The Java Server Pages combine Java language and XHTML tags to produce dynamic web pages. The Java servlets are classes that are written Java language to handle HTTP requests. The servlets use either the POST or GET method to respond to client requests. The Enterprise Java Beans are components developed in Java language. These components include session beans, message-driven beans (MDBs) and entities.

According [5], the session beans can be stateless, stateful, or singleton and they perform business service operations. These services may be declaratively configured to operate in distributed, transactional, and access-controlled contexts. The message-driven beans are invoked asynchronously in response to external events through association with a messaging queue or topic. The Entities are objects that have unique identities and represent persistent business data. They are implemented using the Java Persistence API (JPA).

2.4.1 Implementation Tools

The following tools were used for the implementation in Java:

- NetBeans IDE 7.4 was used for the application development.
- JSP was used for generating the page views.
- Java programming language was used for the controllers (servlets) and models (Enterprise Java Beans).
- MySQL Server 5.5 was used for the database development.
- GlassFish Server 4.0 was used for the web server.

2.5 Implementation in PHP

The system was implemented in PHP using the CakePHP framework. According to [6], CakePHP is a free, open-source and rapid application development framework for PHP. CakePHP templates were used for the views, CakePHP controller classes for the controllers and CakePHP entity classes for the models. The CakePHP templates combine PHP language and HTML tags to produce dynamic web pages. Cake Software Foundation (2014) further describes the CakePHP controllers as classes that are written in PHP language to handle HTTP requests. The entity classes represent database objects and they are responsible for implementing the business logic, which includes processing, validating, associating or other tasks related to handling data.

2.5.1 Implementation Tools

The following tools were used for the implementation in PHP:

- NetBeans IDE 7.4 was used for the application development.
- CakePHP 2.5 was used for generating the page views, controllers and models (entity classes).
- MySQL Server 5.5 was used for the database development.
- Apache Server 2.4 was used for the web server

2.6 Implementation in ASP.NET

The system was implemented in ASP.NET using the ASP.NET MVC 4 framework. ASP.NET MVC is a framework for building web applications that applies the general Model View Controller pattern to the ASP.NET framework. The Razor view engine was used for the views, C# controller classes for the controllers and C# entity classes for the models. The Razor view engine combines C# code and HTML tags to produce dynamic web pages. The controllers were written C# language to process incoming requests, perform operations on the model, and select views to render to the user. The entity classes represent database objects and they are responsible for implementing the business logic [7] – [9].

2.6.1 Implementation Tools

The following tools were used for the implementation in ASP.NET:

- Microsoft Visual Studio 2012 IDE was used for the application development.
- ASP.NET MVC 4 Razor view engine was used for generating the page views
- C# programming language was used for the controllers and models (entity classes)
- Microsoft SQL Server 2012 Express Edition was used for the database development.
- Internet Information Service (IIS 7.0) was used for the web server

V. RESULTS & DISCUSSION

Table 1 compares the average response time of processing Login HTTP requests using each technology. For this comparison, the same application was developed in each of the web development technologies and executed on the same computer, using Windows operating system and Firefox browser. A computer with 2 GB main memory and a 2.17 GHz dual core processor running Windows 7 service pack 1 as the operating system was used. The web browser's network analysis tool was used to measure the response time of processing HTTP requests by each web development technology.

Table 1: Comparison of Web Development Technologies by Average Response Time of Login HTTP Requests

Web Development Technology	Average Response Time of Processing Login HTTP Requests (in milliseconds)	
	GET	POST
Java	32.40	15.60
CakePHP	157.40	162.20
ASP.NET MVC	57.80	62.40

Figure 4 shows a chart of the average response time of processing Login HTTP requests (in milliseconds) for Java, PHP and ASP.NET.

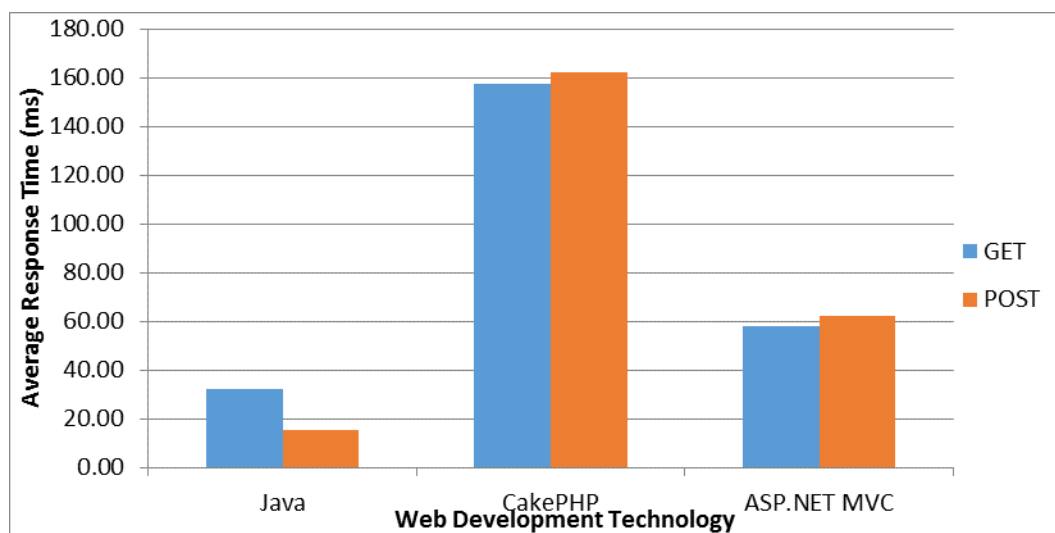


Figure 4: Average Response Time of Processing Login HTTP Requests

The comparison shows that Java processed Login HTTP requests faster than PHP and ASP.NET. The results also indicate that ASP.NET MVC performed better than CakePHP in processing Login HTTP requests. Figure 5 below shows the individual response times at different time intervals for Java, PHP and ASP.NET in processing the Login HTTP requests using the GET method.

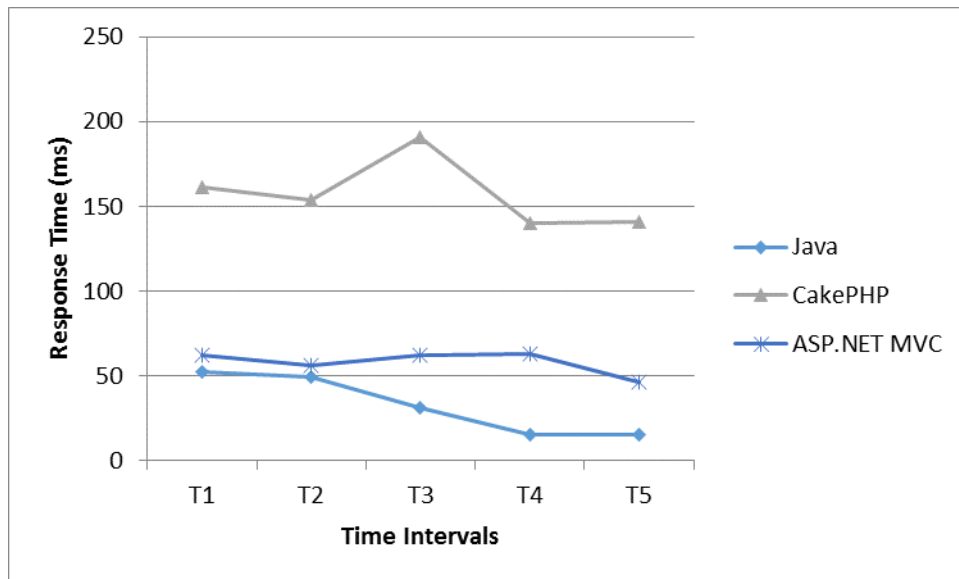


Figure 5: Response Time of processing Login HTTP Requests Using GET Method

These results are in agreement with [2], where it was observed that JSP significantly performed better than Microsoft’s ASP.NET architecture. Figure 6 shows the individual response times of processing the Login HTTP requests using the POST method at different time intervals.

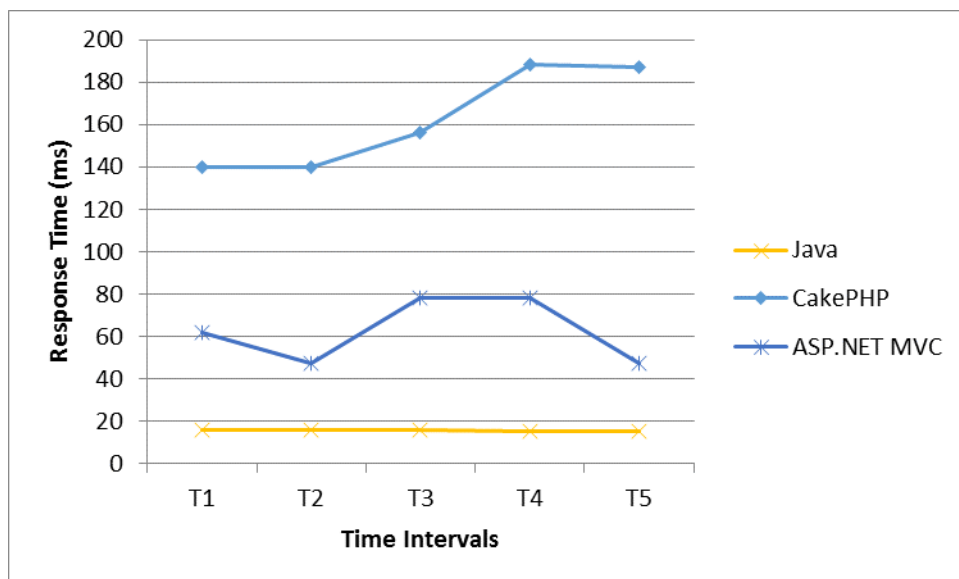


Figure 6: Response Time of processing Login HTTP Requests using POST Method

Table 2 compares the average response time of processing Course Syllabi HTTP requests using each technology.

Table 2: Comparison of Web Development Technologies by Average Response Time of Course Syllabi HTTP Requests

Web Development Technology	Average Response Time of Course Syllabi HTTP Requests (in milliseconds)	
	GET	POST
Java	31.60	222.00
CakePHP	227.80	290.20
ASP.NET MVC	43.60	90.40

Figure 7 shows a chart of the average Response Time for Course Syllabi HTTP Request (in milliseconds) for the Java, PHP and ASP.NET.

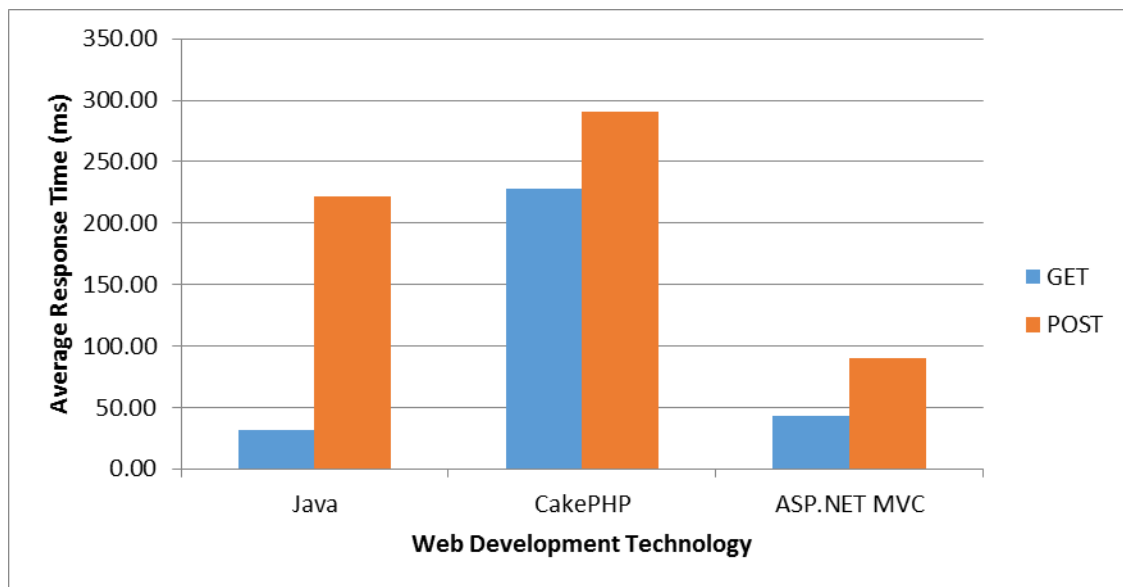


Figure 7: Average Response Time of Processing Course Syllabi HTTP Requests

The comparison shows that Java processed Course Syllabi HTTP requests using the GET method (i.e. listing all course syllabi) faster than PHP and ASP.NET. The results also indicate that ASP.NET MVC performed better than CakePHP in processing Course Syllabi HTTP requests with GET method. However, ASP.NET performed better than both Java and PHP in processing Course Syllabi HTTP requests using the POST method (i.e. saving a course syllabus, including file upload).

Figure 8 below shows the individual response times at different time intervals for Java, PHP and ASP.NET in processing the Course Syllabi HTTP requests using the GET method.

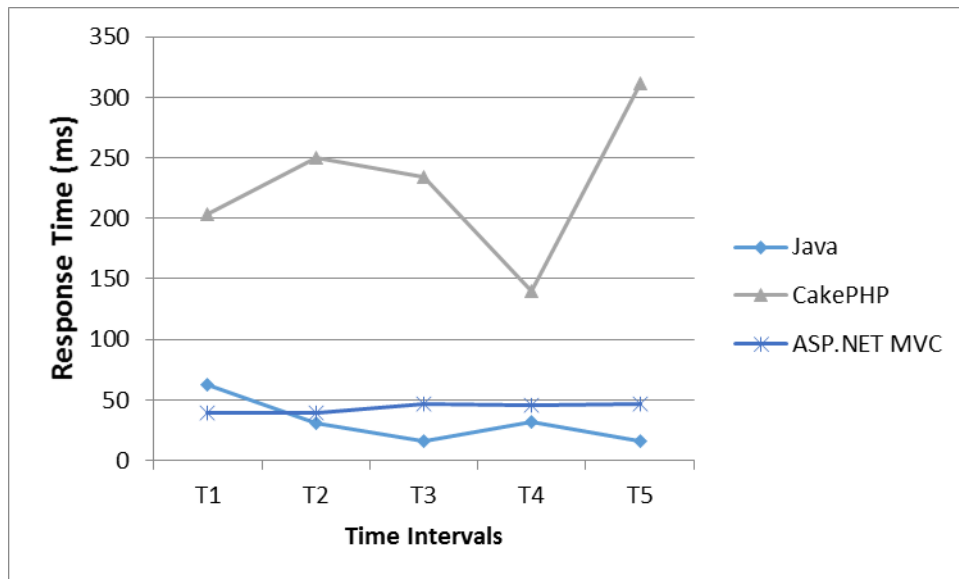


Figure 8: Response Time of processing Course Syllabi HTTP Requests Using GET Method.

Figure 9 shows the individual response times at different time intervals for Java, PHP and ASP.NET in processing the Course Syllabi HTTP requests using the POST method.

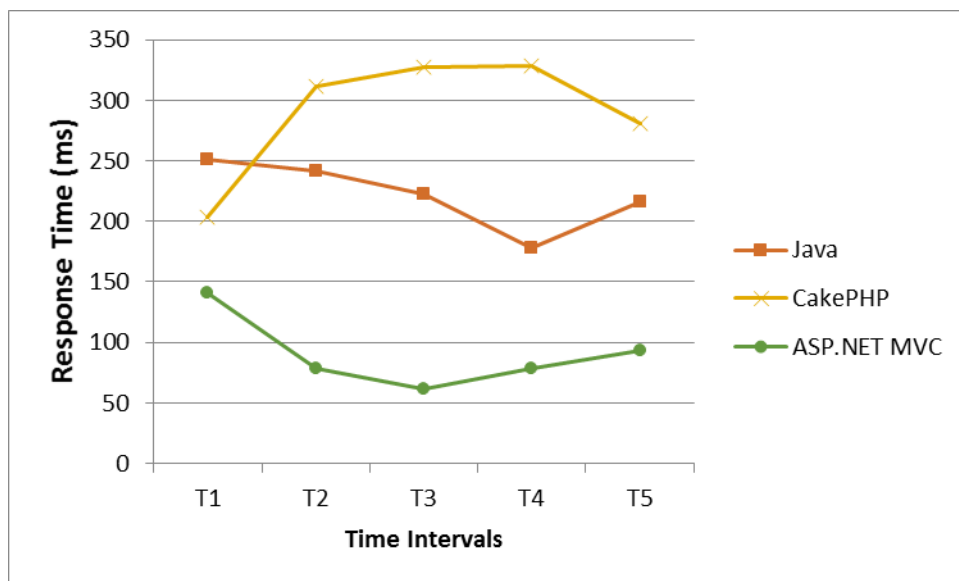


Figure 9: Response Time of processing Course Syllabi HTTP Requests Using POST Method.

Comparing the above results to those of other researchers it was observed that according to [10], PHP worked the best out of the three technologies under consideration in calculating the factorial of 100. However, the use of CakePHP framework in this study showed different results in processing HTTP requests. This can be attributed to the fact that cakePHP is a high-level abstraction of traditional PHP and therefore may not be as fast as the latter. The CakePHP framework was chosen because traditional PHP does not support the MVC design pattern.

VI. CONCLUSIONS

This research has presented a comparison of web application development technologies using open source software and proprietary software. The comparison involved three major web development technologies namely: Java Server Pages (JSP), Active Server Pages (ASP.NET) and PHP Hypertext Preprocessor (PHP). In order to perform the comparison, a web application for submitting examination questions, marking schemes, course syllabuses and examination results, was developed in all the three languages with the same requirements.

The open source software such as Java and PHP were found to be cheaper to implement as compared to proprietary software such as ASP.NET. The study also found out that open source tools such as Java, PHP, Apache Server, GlashFish Server and NetBeans IDE are more compatible with major operating systems for development of web applications than proprietary tools such as ASP.NET, Internet Information Service (IIS) and Visual Studio IDE.

PHP and ASP.NET were found to be more user-friendly as compared to Java. This comparison was based on the model-view-controller (MVC) design pattern that was used in developing the application. ASP.NET MVC and CakePHP frameworks were found to be faster and easier to use than Java. All three technologies proved to be compatible with major web browsers and mobile devices.

Java performed better in processing Login HTTP requests, followed by ASP.NET MVC, as it was observed by measuring the response time of processing HTTP requests on the Login page of the application. In processing Course Syllabi HTTP requests using the GET method (i.e. listing the course syllabi), Java performed better than ASP.NET and CakePHP. However, ASP.NET MVC processed Course Syllabi HTTP requests using the POST method (i.e. saving a course syllabus) faster than Java and PHP. Similar results were obtained in processing requests from other pages. This indicates that in uploading data and files to the server, ASP.NET MVC performs better than Java and PHP. For all other requests, Java performs better than ASP.NET MVC and Cake PHP.

REFERENCES

- [1] Nourie D. (November, 2006), *Java Technologies for Web Applications*, Oracle Technology Network, Retrieved from <http://www.oracle.com/technetwork/articles/javase/ssLINK/142892#nourie>
- [2] Swales D., Sewry D. and Terzoli A (2003), *A Performance Comparison of Web Development Technologies to Distribute Multimedia across an Intranet*, Retrieved from <http://www.satnac.org.za/proceedings/2003/bband/bband2/703 - Swales.pdf>
- [3] Lavanya, R., Ramachandran, V., & Mustafa, J. (2010), *A Comparative Study on Internet Application Development Tools*, International Journal of Engineering Science and Technology, 2 (10), 5452-5456.
- [4] Leff, A., & Rayfield, J.T. (2001), *Web-Application Development Using the Model-View-Controller Design Pattern*, Paper presented at the Fifth IEEE International Enterprise Distributed Object Computing Conference, 118-127.
- [5] Wetherbee J., Rathod C., Kodali R. and Zadrozny P. (2013), *Beginning EJB 3 Java EE 7 Edition*, Apress, 25-30.
- [6] Cake Software Foundation (2014), *CakePHP Overview*, CakePHP Cookbook Documentation, 45-50.
- [7] Galloway, J., Haack, P., Wilson, P., & Allen K.S. (2012), *Professional ASP.NET MVC 4*, J. Wiley & Sons, 2-8.

- [8] Feeman, A. (2012), *Pro ASP.NET MVC 4*, Apress, 47-51.
- [9] Palermo J., Bogard J., Hexter E., Hinze M. and Skinner J. (2012), *ASP.NET MVC 4 in Action*, Manning Publishing Co., 5-9.
- [10] Ranjan A., Kumar R. and Dhar J. (2010), *A Comparative Study between Dynamic Web Scripting Languages*, Proceedings of ICDEM, 288-295.