Visual PHP

By

Mohd Faeiz bin Mohd Taib

Dissertation submitted in partial fulfillment of
The requirements for the
Bachelor of Technology (Hons)
(Business Information Technology)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

Visual PHP

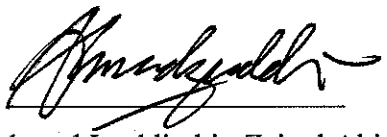
By

Mohd Faeiz bin Mohd Taib

A project dissertation submitted to the
Information Technology Programme

Universiti Teknologi PETRONAS
in partial fulfillment of the requirement for the
BACHELOR OF TECHONOLGY (Hons)
(BUSINESS INFORMATION TECHNOLOGY)

Approved by,



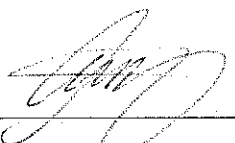
(Mr. Ahmad Izuddin bin Zainal Abidin)

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2006

CERTIFICATE OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work in my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



MOHD FAEIZ BIN MOHD TAIB

Acknowledgement

It is increasingly difficult to acknowledge all the people who have influenced in this project. Lecturers, classmate and friends have contributed their knowledge and insight, often without knowing their impact. By arguing a point or sharing explanations of concepts, it somehow creates a solution to a problem or creates and insight to include in the project. I also would like to express my appreciation for the excellent support received from my supervisor Mr. Ahmad Izuddin bin Zainal Abidin.

I am very grateful for the received support and outstanding contribution from the mentioned people which have helped in the development of this project.

Abstract

Generally What You See Is What You Get (WYSIWYG) editor is very popular among all programmers in the world. In this project the editor is built specifically for PHP programming language. The project was inspired by the booming of PHP language user and also the needs for the editor itself. From day to day the PHP programmers regardless of novice or expert are looking for perfection in the editor which can give the best features and help to them in writing the PHP application. Therefore this project was initiated purposely to cater the needs for some segments of PHP programmer who likes the PHP to be coded in Visual Programming environment. So the project has been applying the Model-View-Controller (MVC) to achieve the objective in providing the visual programming environment. Other than that, the project also includes some helps and tips to the programmer as well as the intelligent aspects to the system for example Intelli Sense®. As a result, the project is able provide the user with fully featured PHP visual editor.

TABLE OF CONTENTS

TABLE OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Background of Study

PHP (PHP Hypertext Preprocessor) has come a long way in the last few years. PHP is one of the most popular server sides scripting available for public under free software license. In fact, PHP language is very powerful and popular among web programmers in developing database driven application. This programming language requires other software for web server and database which is also available at free software license or in other words an open source.

This condition has led a rapid growth in PHP to be one of the most prominent languages powering the. The PHP usage over the internet also increases significantly over the years. There is a statistic done by Netcraft.com to prove the increment in the usage of PHP as web base scripting.

Jsage Stats for April 2006

HP: 20,475,056 Domains, 1,278,828 IP Addresses
ource: Netcraft

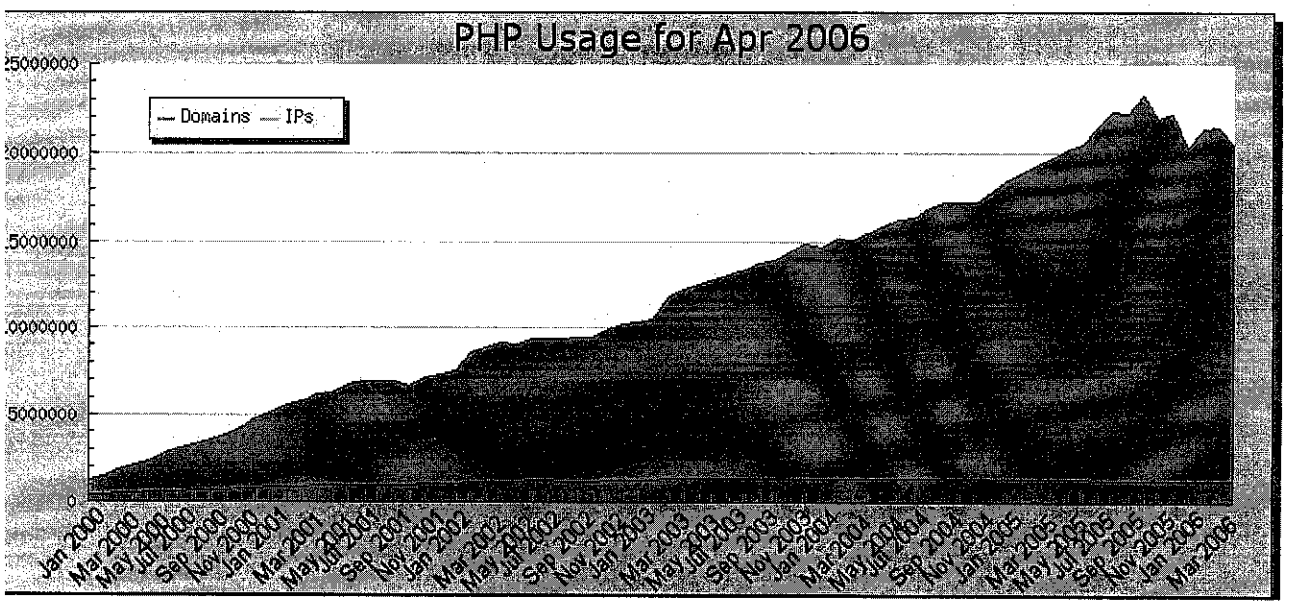


Figure 1.1: PHP usage statistic

Position May 2006	Position May 2005	Delta in Position	Programming Language	Ratings May 2006	Delta May 2005	Status
1	2	↑	Java	21.316%	+3.92%	A
2	1	↓	C	17.690%	-0.83%	A
3	4	↑	C++	10.935%	+1.03%	A
4	5	↑	PHP	10.220%	+1.02%	A
5	6	↑	(Visual) Basic	9.564%	+1.93%	A
6	3	↓↓↓	Perl	6.050%	-4.53%	A
7	7		C#	3.271%	+0.50%	A
8	8		Python	3.042%	+0.56%	A
9	11	↑↑	JavaScript	2.191%	+0.75%	A
10	9	↓	Delphi/Kylix	1.766%	-0.48%	A
11	17	↑	SAS	1.479%	+0.14%	A

Figure 1.2: TIOBE Programming Community Index for May 2006

The TIOBE Programming Community index gives an indication of the popularity of programming languages. The index is updated once a month. The ratings are based on the world-wide availability of skilled engineers, courses and third party vendors. The popular search engines Google, MSN, and Yahoo! are used to calculate the ratings. Observe that the TPC index is not about the best programming language or the language in which most lines of code have been written. The index can be used to check whether your programming skills are still up to date or to make a strategic decision about what programming language should be adopted when starting to build a new software system.

PHP has become more mature and popular programming language as time goes on. As one of PHP programmer, I see the needs of most PHP programmers in the world. Most of them are looking for the most perfect PHP editors or WYSIWYG editor. Therefore I see the opportunity where they are looking for visual developing environment as well as editor which integrate with more intelligent aspect. This condition and demand

has contributed to an idea or insight to me to create an editor which consists of those two criteria.

The visual developing aspect can be explained as visual programming language. Visual programming language (VPL) is any programming language that lets users specify programs in a two-(or more)-dimensional way. Conventional textual languages are not considered two-dimensional since the compiler or interpreter processes them as one-dimensional streams of characters. A VPL allows programming with visual expressions, spatial arrangements of text and graphic symbols. Non Visual Programming languages are completely based on text, when using non VPL applications it is impossible to see images until the program are run. This is a disadvantage to programmers as the user is unable to see the results of their coding unless the program is run.

Intelligent aspect is an artificial intelligence area which also can be applied in code editor which most of programmers are looking for. Intelligent aspect in code editor is important as it creates a convenient way or environment to the coders so that they can write codes at a faster rate and at minimized spelling errors. This condition can be achieved by providing such intelligent agent for example IntelliSense™, code completion and more. These two features visual programming environment and intelligent aspect will be explained in detail in later chapter.

1.2 Problem Statement

1.2.1 Problem Identification

Those objectives are also derived from the problems and difficulties faced by the programmer in programming PHP codes. There are several circumstances that create the opportunity to develop new IDE for PHP. Those circumstances are:

- Programmers are prone to perform typos in less guided text editor software.
- Writing PHP application codes sometimes can drag programmers for a longer period when there are several mistakes in the codes.
- PHP language newbie are not confidence writing PHP codes from scratch.
- Some programmers write PHP codes in a way that reduce the readability of the codes which could to many difficulties for future enhancement of the application

1.2.2 Significant/Benefits of This Project

After further research there is some changes in terms of the system features and at the same time will alter the significance of my project. The changes are considered based on the limitation of time and resources in developing this project.

- Rich of features
  - Auto complete
    - This feature is possible to be provided in the tool, but the only thing is that much time is needed in order to study on how it can be done using VB 6.0
  - Internal http server
    - Most probably this feature is going to be taken out of the software since the requirement for it is not too strict. It only needs at least an internal web browser so that user can test their application assuming that PHP is installed on the machine.
  - Auto generate code
    - This function will remain in the application but there is an addition to this feature which a user customizability on the generated output

- Support wide range of users.
  - Since this project provide both GUI based and Text based programming approach, it is suitable to all level of programmer, from newbie to advanced programmer.

1.3 Objective and Scope of Study

1.3.1 Main objectives of this research/project:

The need of PHP IDE (Integrated Development Environment) has increase as the increase of the PHP programmers over the years. This also has encouraged me to develop a program that will help PHP programmers. The purpose of the Visual PHP project is to ease and speed up the development of PHP application.

The main objective of this program is to build complete IDE software that combines all the features available third party PHP application into fully featured software. Besides the major objective there are several other objectives lie behind the IDE:

- To minimize possible error in typing PHP codes
- To guide newbie writing PHP codes without require them to memorize PHP functions
- To rapid the PHP application development process

1.3.2 Scope of Study

Upon completion of the project it is understood that most of PHP well enjoy the benefits. Since it provides both text based and GUI base, therefore it supports wide range of PHP programmers. The software produced also will base on PHP version 4 and 5 which will give opportunity to experience programmer in PHP version 4 to take a closer look at PHP version 5. The other part of the program is the database, it only cover MySQL database since MySQL database is very popular among PHP programmers, besides the good integrity among MySQL and PHP. This program also will support only simple database manipulation function. This is a web base program therefore it only support programming in PHP and XHTML. Which means it doesn't assist programming in other language.

Environment of the project is developed and tested on windows platform, therefore it is 100% compatibility with windows OS. The requirement of the project may change as the development going on. The limitation of knowledge and time may somehow alter the design of the project. But it is hope that this project would end up successfully and reliable to the user.

1.3.3 Comparative Study

Besides doing the literature review I also had done the comparative study between the available software on the market. The study had given me a good impact in the analysis of my project. From the comparison it creates a broader view in developing the project. The study also gives me a bigger view of the scope of the project on the features that are going to be included in the program.

There are several commercial software available on the internet nowadays for example:

- Web-based PHP Class Generator - A very basic PHP class builder to generate class templates useful for object-oriented programming in PHP.
- MySQL PHP Code Generator - Generate all the PHP scripts need to manage in PHP data from MySQL database
- EngInSite Editor for PHP - A complete integrated development environment (IDE) for creating, testing and debugging PHP script

Basically, those available programming tools do not comes with fully featured software. What I meant by fully featured software is some tools only provide the database manipulation and some only provide the IDE. These tools can be enhanced by combining the database manipulation function with IDE to provide the faster development kit which at the same time provides the coding flexibility. By virtue of flexibility in programming and the speed of the development both GUI based and text based has to be in one program in order to suit with the needs of the programmer and the project going to be developed. After preliminary report is submitted, there is a further and thorough study done to the available software. This study is actually part of the research in determining the system requirement.

CHAPTER 2

THEORY

2.1 Literature Review

2.1.1 Why another IDE?

Much in the same way as with BlueJ, we saw the need for a lightweight, easy-to-use teaching environment for the C# language. The Visual Studio .NET IDE has been developed over many years and is the de facto standard for application development on the Microsoft Windows platform. The main problem from a teaching viewpoint (now that licensing ceases to be such an issue) is that it is complex and daunting for students. Once the initial complexity is learnt then the functionality enables the student to be much more productive (and adds another "skill" to the CV), but as with most tools, it often masks the underlying concepts. So why not just produce a "clone" of the BlueJ environment for another language, i.e., C#? After all, there has been widespread acceptance of BlueJ

4• Over many years of teaching object-oriented **programming**, we have become aware of the visual metaphors that students find most helpful for learning object-oriented concepts. For this reason we believe the environment should display the class structure visually5. • When teaching object-oriented design and **programming**, we spend most of the time discussing class structure. Students typically experience difficulty thinking in terms of classes and objects (especially if they have previous experience in using procedural-**programming** languages); yet structuring the problem via these concepts is one of the most important aspects of object-oriented **programming**. • The visualisation process is made much easier if students do not just see lines of **code** but also have a design view of the software they are developing (cf. Rational Rose 6). These are all desirable aspects of an IDE system, but some of the drawbacks we have noted with

BlueJ-like systems are: • that the knowledge of using a specific IDE with reduced functionality does not necessarily transfer to a more feature-rich environment. Our students start off with BlueJ for their Java **programming**, but have to migrate to

NetBeans (Sun ONE Studio) 7 or a similar IDE for more complex **programming** work (e.g., J2EE, work, etc). This requires explicit instruction to support the move and it is often necessary to supplement this with appropriate exercises; • that it may be reasonably easy for students to tackle exercises in which they have to modify existing source **code** (facilitated by the **programming** environment's **code** templates and stubs), but it is not an easy transition for them to write classes from scratch. This can lead to wasted time and sometimes complete confusion for the student. This also occurs if they have to work in a command shell environment; what they have learnt does not easily transfer. Simple things like constructing an object often cause the students many syntactic and semantic problems [1].

2.1.2 Literature Review 2 (The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency)

An experimental GUI paradigm is presented which is based on the design goals of maximizing the amount of screen used for application data, reducing the amount that the UI diverts visual attentions from the application data, and increasing the quality of input. In pursuit of these goals, we integrated the non-standard UI technologies of multi-sensor tablets, toolglass, transparent UI components, and marking menus. We describe a working prototype of our new paradigm, the rationale behind it and our experiences introducing it into an existing application. Finally, we present some of the lessons learned: prototypes are useful to break the barriers imposed by conventional GUI design and some of their ideas can still be retrofitted seamlessly into products. Furthermore, the added functionality is not measured only in terms of user performance, but also by the quality of interaction, which allows artists to create new graphic vocabularies and graphic styles.

The basic components of a GUI reflect the characteristics or subtasks of a user's workflow. For example, in the drawing domain, original interfaces like MacPaint have UI components like a tool palette and scrollable drawing surface. This roughly reflects the way an artist would work with pencils and paper. The user moves between the palette and drawing surface, drawing and changing their focus of attention (navigating) to different portions of the drawing. Since selection from the tool palette, drawing and navigation are frequent tasks; GUI designers make these functions readily accessible generally by constantly displaying the UI widgets for these functions. **While this design** approach has **been** very successful it does create some design tensions. First, it introduces a competition for screen space between the UI widgets and the user's
Art work (Figure I). Second, it produces a dichotomy between UI widgets and the artwork where a large majority of the UI widgets exist around the edge of the artwork. The first design tension could be addressed by a larger screen with the cost being the expense of a larger display. However, as screen size increases the second design tension becomes a problem. As the screen and artwork become larger the distance a user must travel to/from a tool palette or menu increases. This results in longer task times. Furthermore, a user's focus of attention must constantly change from some point on the

artwork to a UI widget at the edge of the screen and then refocus on the artwork again. Dividing attention in this manner requires additional time to reacquire the context and can also result in users missing some message from the system or some change in the artwork performed by the system. We believe that this divided attention problem significantly affects the quality of a user's interaction 1. In addition to these design goals we also wanted to address the issue of the quality of Input in a traditional GUI. Original GUIS such as the Xerox Star and the Macintosh assumed the mouse and keyboard to be the basic input devices. A huge amount of the power of the traditional GUI comes from the fact that the mouse allows continuous 2 dimensional inputs from one of the user's hands. We were interested in how providing continuous input for the other hand would improve or affect the design of a GUI.

In this paper we describe an experimental GUI which attempts to address these issues. We designed a GUI paradigm (model of interaction) with the following design goals:

Artwork: Maximize the amount of screen used for artwork

Focus: Avoid forcing the user to divert their visual attention from the artwork

Input: Increase the degrees of manipulation and comfort of input.

2.1.3 INTELLIGENT ADAPTIVE SYSTEMS

Intelligent systems are able to act rationally to seek optimal solutions for their design objectives. From the user perspective, this means delivering context-aware human-like services with minimal user effort. Often, what seems rational to one user is not rational to another. Therefore, the system may need to reason about and adapt to individual users and modify its behaviour accordingly. As a natural extension of intelligence, such systems should seamlessly integrate with other applications that have not been explicitly designed to work together. In the example of a Smart Home Network, the system should provide customized services that are rational to the actual family it's serving, and learn to accommodate their particular needs.

2.1.4 Usability Analysis of Visual Programming Environments: a 'cognitive dimensions' framework

Psychology and HCI of Programming

The framework of 'cognitively-relevant dimensions' is founded (not too tightly, alas) on present-day views on the activity of programming, which we shall review very briefly. 'Programming' is a seriously over- loaded term, comprising a host of different activities and situations [61]; perhaps it is fortunate that we have no space to enter into the niceties here. What follows is a very high-level view, restricted to that which is relevant to the cognitive dimensions framework.

To start with, we need to consider the users and the situation. If we wish to design a programming environment, who is it for? What are they doing? What aspects of the activity are affected by the programming environment?

First, we shall assume that anyone may be a user; expert or novice, end-user or computer-science professional. We shall ignore the large literature on what experts know that novices do not know, but we shall take into account the extra support that novices need. Second, we shall tend to limit our discussion to situations like exploratory or incremental programming. We shall pay no attention to other design criteria, such as safety-critical design or coding for efficiency; nor to other parts of the software creation process, such as communication and negotiation during requirements elicitation; nor to the demands of the local situation or organization, even though they are known to affect choice of cognitive strategy [87]. One thing at a time! And thirdly, within those limits we shall try to consider as much of the programming process as is affected by the programming environment. Not just coding, or just comprehension. We shall distinguish rather loosely between the 'psychology' and the 'HCI' of the programming environment, using 'psychology' to refer to the meaning of the code ("How do I solve this problem? What does that code mean?") and 'HCI' to mean interaction with the notational system (control of layout, searching for items).

2.1.5 Advantages of Visual Programming Languages

Many authors argue that VP systems are a better method for users to interact with a program. Green et. al. [Green *et al.*, 1991] and Moher et.al. [Moher *et al.*, 1993] summarise claims such the above quote from [Hirakawa & Ichikawa, 1994] as *the superlativist position*; i.e. graphical representations are inherently superior to textual representations. Both the Green and Moher groups argue that this claim is not supported by the available experimental evidence. Further, they argue against claims that visual expressions offer a higher *information accessibility*; for example:

Pictures are superior to texts in a sense that they are abstract, instantly comprehensible, and universal. [Hirakawa & Ichikawa, 1994]

My own experience with students using visual systems is that the visual environment is very motivating to students. Others have had the same experience:

The authors report on the first in a series of experiments designed to test the effectiveness of visual programming for instruction in subject-matter concepts. Their general approach is to have the students construct models using icons and then execute these models. In this case, they used a series of visual labs for computer architecture. The test subjects were undergraduate computer science majors. The experimental group performed the visual labs; the control group did not. The experimental group showed a positive increase in attitude toward instructional labs and a positive correlation between attitude towards labs and test performance. [Williams *et al.*, 1993]

For another example of first year students being motivated by a VP language, see [Glinert & Tanimoto, 1984] (p18-19). However, merely motivating the students is only half the task of an educator. Apart from motivating the students, educators also need to train students in the general concepts that can be applied in different circumstances. The crucial case for evaluating VP systems is that VP systems improve or simplify the task of comprehending some conceptual aspect of a program. If we extend the concept of VP systems to diagrammatic reasoning in general, then we can make a case that VP has some such benefits. Larkin and Simon [Larkin & Simon, 1987] distinguish between:

- *Sentential representations* whose contents are stored in a fixed sequence; e.g. propositions in a text.
- *Diagrammatic representations* whose contents are indexed by their position on a 2-D plane.

While these two representations may contain the same information, their computational efficiency may be different. Larkin and Simon present a range of problems modeled in a diagrammatic and sentential representation using production rules. Several effects were noted:

- *Perceptual ease:* Certain features are more easily extracted from diagrams than from sentential representations. For example, adjacent triangles are easy to find visually, but require a potentially elaborate search through a sentential representation.
- *Locality aids search:* Diagrams can group together related concepts. Diagrammatic inference can use the information in the near area of the current focus to solve current problems. Sentential representations may store related items in separate areas, thus requiring extensive search to link concepts.

2.2 Conclusion of literature Review

The research paper and journal reviewed has contributed in creating new ideas to do my project and to stipulate some objectives of the project. Basically, the review also has given me such a great support or evidence to proof that this project is a good project and should be continue until it is finish.

CHAPTER 3

METHODOLOGY

3 Procedure Identification

3.1 System Development Process Model

Method that is used in developing the system is the waterfall model. Waterfall model is one of the most popular incremental project development methodologies. The development of the project conducted by sequence starting with analysis, design, develop, test, and lastly implementation. I choose the waterfall model because the concept is applicable to my project. A thorough analysis need to be done before I can proceed further. But, if anything happen in the future to the development of the project, based on this model I can still go back to the previous stage to correct of the analysis information so that it is relative with the output of the project.

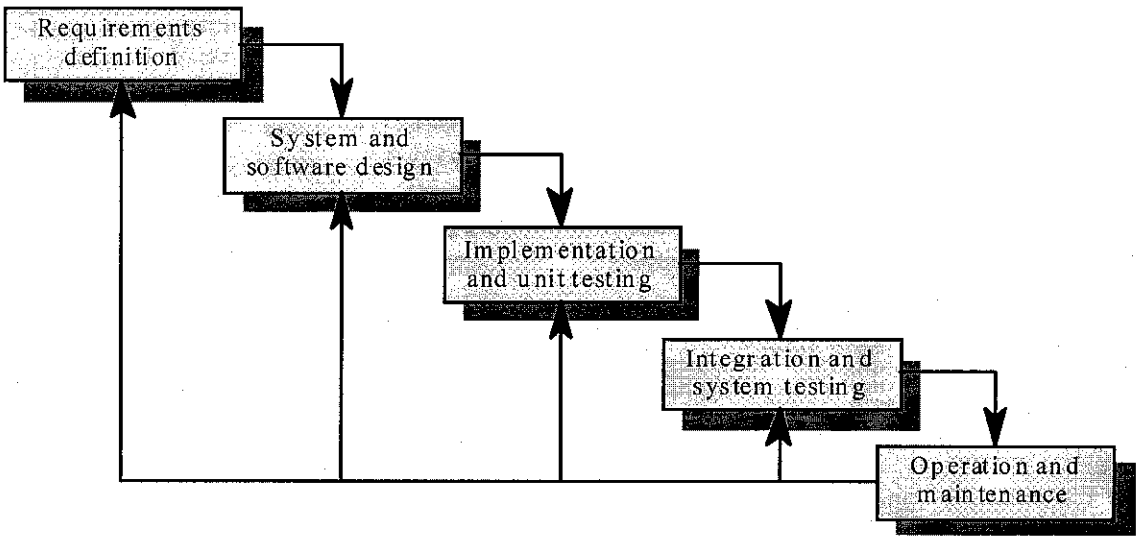


Figure 3.1: Waterfall model (software development methodology)

Waterfall model is popular method because many programmers like to code a program from the approved requirement and thorough analysis. Good preparation before kicking off the coding part is a good practice because in this methodology it is quite hard to go back a step behind, even though it can be done. But, refining the requirement after starting the development stage may take some times to correct or change the prototype. But anyway, waterfall model is a great choice if all the requirements are clear.

3.1.1 Research, Analysis and Review

The goal of the Analysis phase is to capture all of the user specifications for the project and to completely detail all business processes that will be involved. This is the most important step in systems design. I have broken the Analysis phase into two parts: Information Gathering and Requirements Analysis.

a. Information Gathering
    i. Literature Review is part of information gathering in order to get the concept and know how the implementation of certain concept for example user friendliness of a program.
    ii. This stage studies the feature or the characteristic that the system should be.
    iii. This stage is where the study on how the feature of the system can be implements in the project.
    iv. Study about VB
        1. Study about VB 6.0, on how to create certain function in order to complete the project.
        2. There is some limitation in term of knowledge of VB 6.0 that drags the development time become longer.

v. CodeMax component version 4

1. Codemax is VB6 control which provides the programmer with the control to create a text editor or programming code editor.
2. The codemax control has very less documentation or help file which has require much time to learn and study how to use the control in the project.

    a.

b. Requirement Analysis

i. Comparative study between the currently available software tools in order to get the functional specification.

ii. Requirement specification is developed at this stage a guideline for functional specification of the application.

3.1.2 System Design

In the phase between Analysis and Design, the rest of the project is planned. The Pre-Design phase includes the following: Design Plan, process flow (physical), design standards, screen concept prototype. After the pre-design is the design phase. This is where the blue print of the project is line out and then based on the blueprint the project is created.

3.1.2.1 System Architecture

System architecture of a program or computing system is the structure or structures of the system, which comprise software elements, the externally visible properties of those elements, and the relationships among them. System architecture forms the backbone for any successful software-intensive system. Architecture is the primary carrier of a software system's quality attributes such as performance or reliability. The right architecture - correctly designed to meet its quality attribute requirements, clearly documented, and carefully evaluated - is the key player for software project success. The wrong one is a recipe for guaranteed disaster.

Basically the system architecture of the project is based on three main components. They are:

1. Vb components
    a. Codemax 2.0
    b. DhtmlEdit
    c. PictureBox

2. Backend codes create the objects at runtime (DHTMLEDIT).

3. Object generate codes in editor(Codemax)

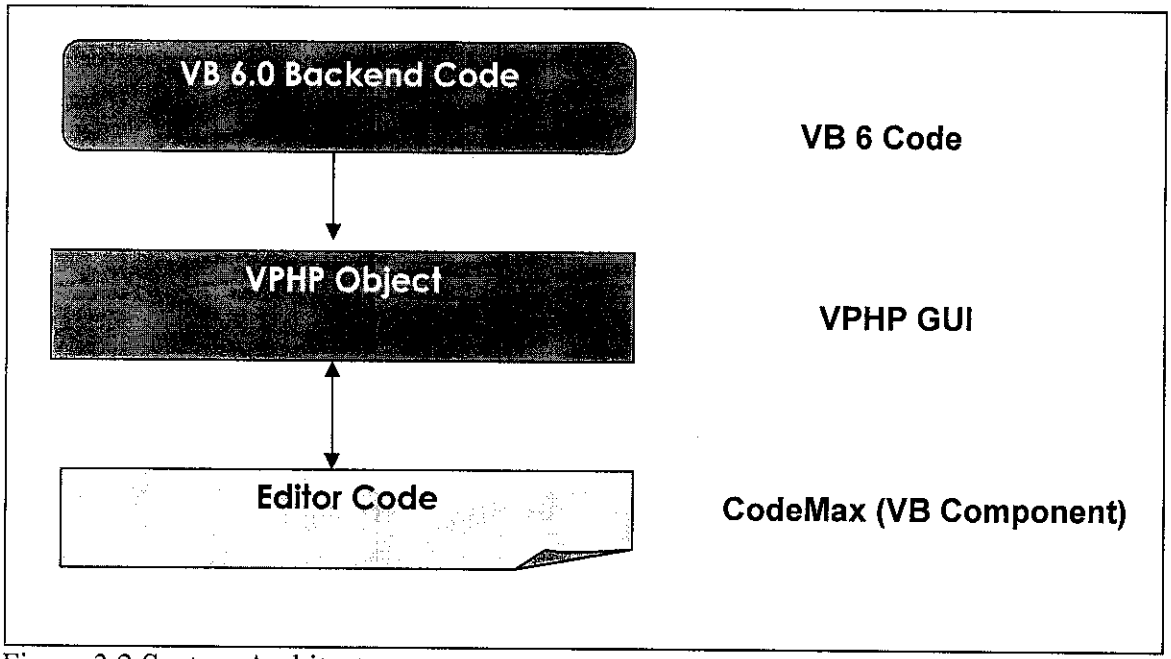


Figure 3.2 System Architecture

Model-View-Controller

One of the most popular system architecture is the Model-View-Controller (MVC). MVC is often thought of as a software design pattern. However, MVC encompasses more of the architecture of an application than is typical for a design pattern. MVC is a software architecture that separates an application's data model, user interface and control logic into 3 distinct components so that modification to one component can be made with minimal impact to the others.

This project was developed based on the MVC architecture.

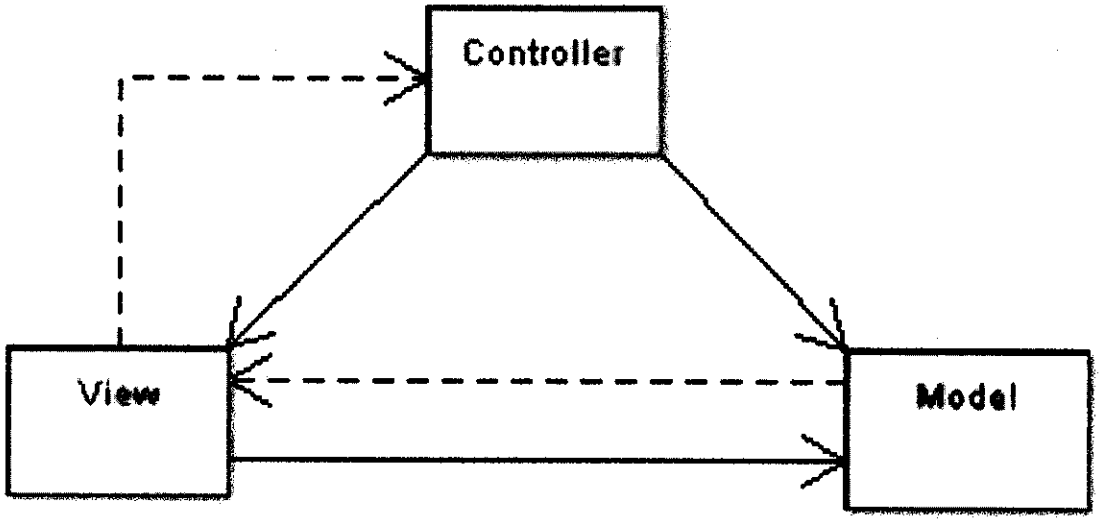


Figure 3.3: Model View Controller model

In order to use the MVC architecture, several things need to be defined they are there classes of modules:

1. Model
    a. The domain-specific representation of the information on which the application operates
    b. In the case of this project, the model would be the object of PHP application for example the button, form, table, functions and, etc.

2. View
    a. Renders the model into a form suitable for interaction, typically a user interface element. MVC is often seen in web applications, where the view is the HTML page and the code which gathers dynamic data for the page.
3. Controller
    a. Responds to events, typically user actions, and invokes changes on the model and perhaps the view. So to relate the project, controller is the one that response to any changes made to the model or objects and results in the changes in the view as well and vice versa.

3.1.3 System Development

Generally, in this stage the product is turn from the design into the output observable on the computer screen. But in this project, the previous stages also involve sometimes in the development phase. This is because during the research and development a lot of information is gathered which sometimes affect the system. In fact, there is a case where the significant features of the project need to be refined and justified because as time goes on some limitation are found in the project.

The development phase is more or less a research and development phase because a lot of study needs to be done to the available software and also a lot of Visual Basic 6.0 function and tutorial need to be understood in order to develop the main part of the project. The main part of the project is the design view, which is the system requirement of the Visual PHP project.

The progress of the Visual PHP project can be divided into two which are the major part and minor part of the project.

Visual PHP Project:

1. *Major Part* is the part which involves the design view of the system. Basically this part is called as the major part is because it is the key feature to prove the significant of this project. The design view that I have been planning to do is more or less the Visual Basic 6.0 design view. PHP programmer will be able to drag and drop some object onto a form, and then the system will automatically generate the source code.

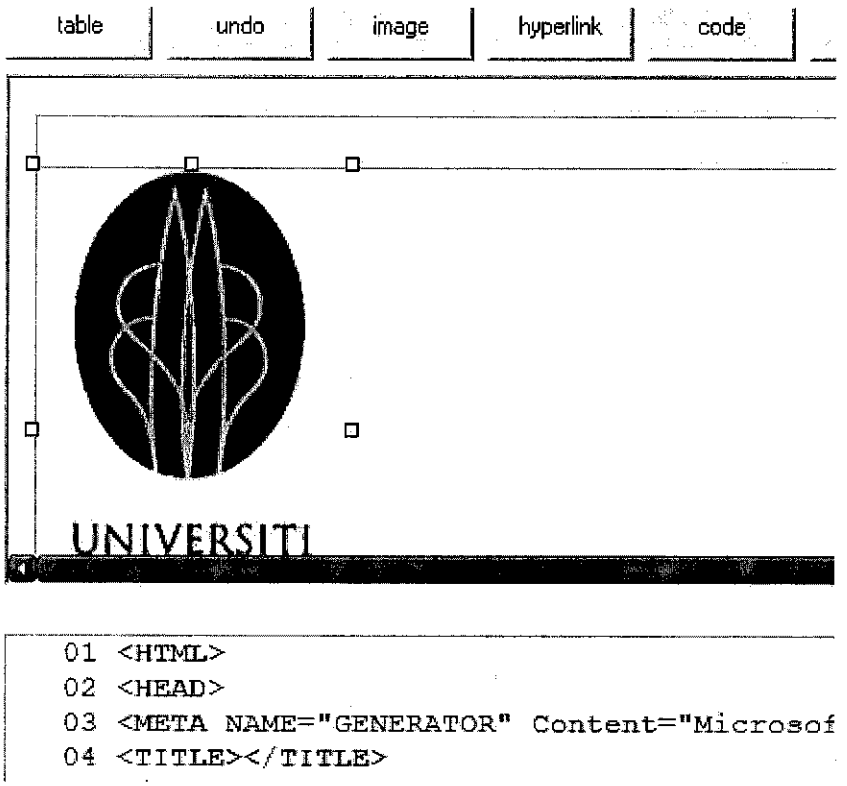


Figure 3.4: Design View

2. *Minor Part* is the part in which involve the **coding editor** and the overall **interface design** of the project. These two parts are categorized as minor part because it is easier to develop compare to the major part. Generally, this project is developed according to the features of the program. So, basically most of the features are ready or finish but still need some improvement so

that it will be a good product. So basically these are some screen shot of the product according to its feature.

Coloured Syntax

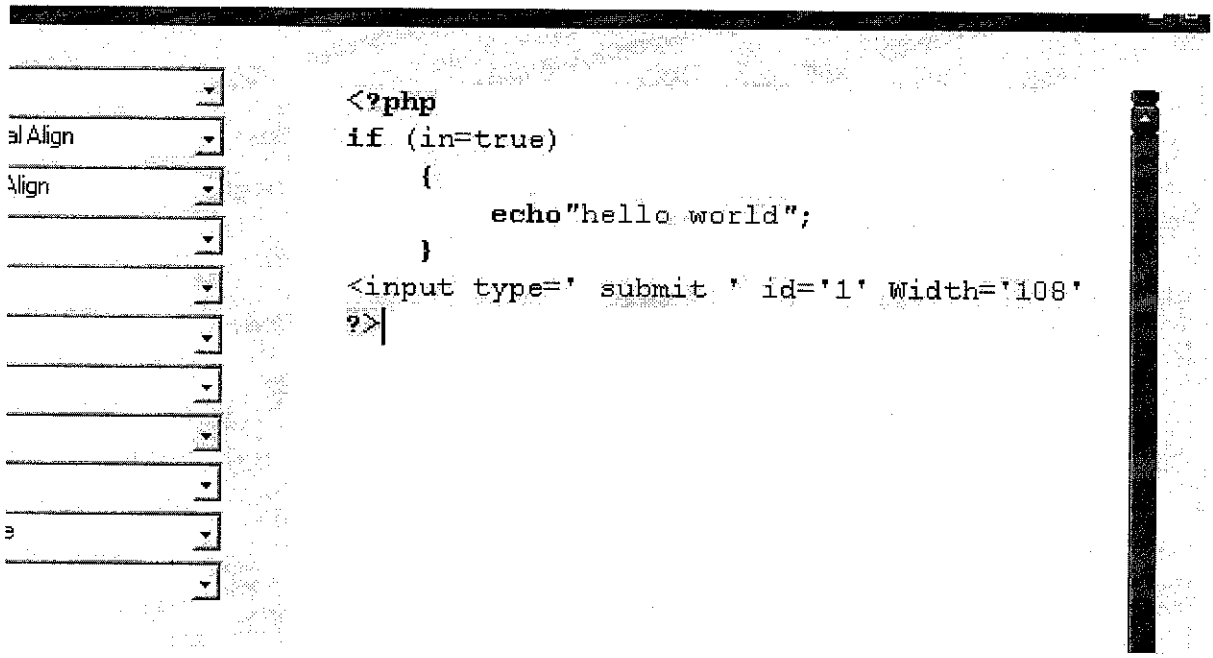


Figure 3.5: Coloured Syntax

Code list

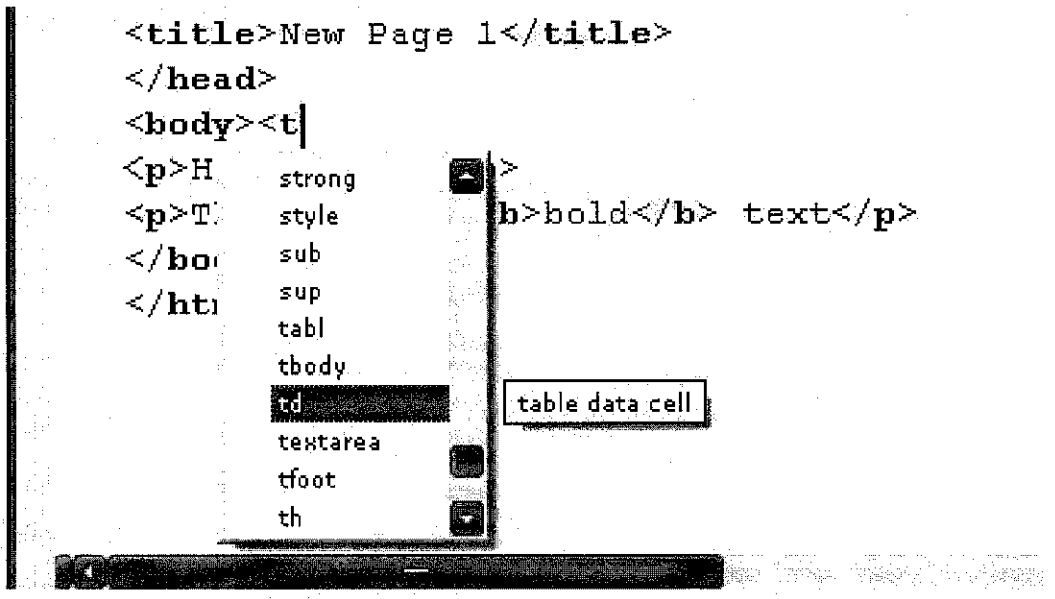


Figure 3.6: Code List

Code tips

Figure 3.7: Code Tips

Actually, the progress of the project development is quite slow due to some problems and limitations. The limitation of Visual Basic 6.0 language knowledge is one of the limitations which I have to face. Lack of references also creates a problem which also drags the development process to a longer time. However, there are some improvements in terms of the interface design and the coding editor parts. For example, certain colors are applied to certain command or code in the coding editor. This function is very important in order to differentiate between command and data which will increase the level of readability of the codes.

3.1.4 System Testing

In software development, testing is usually required before release to the general public. This phase of development is known as the alpha phase. Testing during this phase is known as alpha testing. In this project the alpha testing is done to the prototype of the project. There several steps or procedures that need to be follow in order to get a good result from the test.

3.1.4.1 Identify the Tester

Before testing can be started, first the correct testers need to be identified so that correct result can be derived. Therefore for the sake of this project, to identify the testers I have looked back to the scope of the project, who are the target user of the program. So basically, there are two different categories of user who are the novice and intermediate user. Therefore, for the testing purposes 5 persons from each category has been identified to undergo the testing.

3.1.4.2 Identify the Testing Method

Think aloud protocol is a method used to gather data in usability testing in product design and development, in psychology and a range of social sciences. Think aloud (also spelled think-aloud) protocols involve participants thinking aloud as they are performing a set of specified tasks. Users are asked to say whatever they are looking at, thinking, doing, and feeling, as they go about their task. This kind of testing method gives the observer opportunity to communicate directly with the testers to get the feedback from them.

3.1.4.3 Identify Testing Area

3.1.4.3.1 Usability testing

Usability testing is a means for measuring how well people can use some human-made object (such as a web page, a computer interface, a document, or a device) for its intended purpose, for example usability testing measures the usability of the object. If usability testing uncovers difficulties, such as people having difficulty understanding instructions, manipulating parts, or interpreting feedback, then the application will be redesign and test it again. During usability testing, the aim is to observe people using the product in as realistic a situation as possible, to discover errors and areas of improvement. Besides that, during the test there are also criteria that were looked into whether the application meet the objectives stated in the previous section.

3.1.4.3.2 Product Comparison

Comparison testing is a test in which the objective is to see effectiveness and accuracy of a specific application compared to other applications.

3.2Tools Used

1. VB 6.0– as the main programming language
2. Apache – as the web server to test the PHP scripts
3. MySQL – as the database for PHP scripts
4. PHP –
5. Adobe Photoshop 7 – as the design tool to design graphics or images.

CHAPTER 4

RESULTS AND DISCUSSION

4. Testing Result

4.1 Result for Usability testing

How long the user took to understand and use the program?

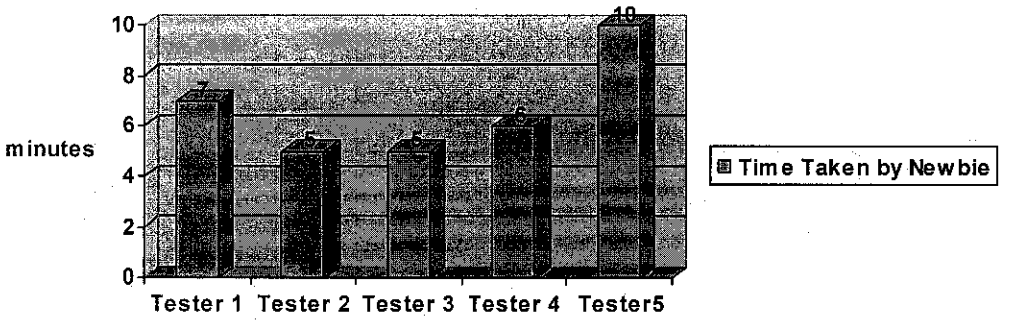


Figure 4.1: Usability Test Result (1a)

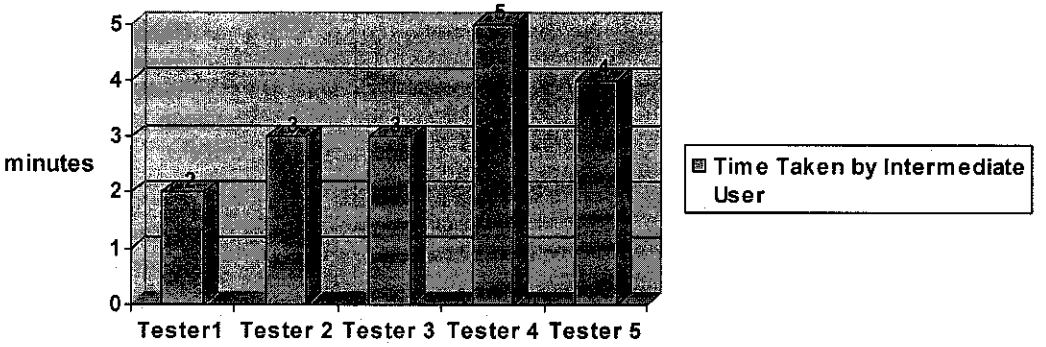


Figure 4.2: Usability Test Result (1b)

The results show that, the average time taken by each user to understand the user interface and use the program for about 4.9 minutes. From the result and feedback from the tester itself the interface is good but any improvement is on graphics can increase the user friendliness of the program.

What does the user feel about the system?

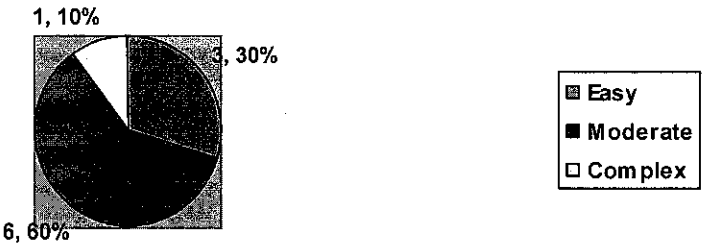


Figure 4.3: Usability Test Result 2

From the result and feed back from the testers, it shows that many say that the system complexity is in between easy and complex. Therefore, it is concluded that the current user interface is good. Because those tester with more experience they said that the system is easy to use because it uses the same metaphor as other programming editors. Basically, the program complexity is moderate based on the majority of the tester.

Does the program helps in typing the PHP codes?

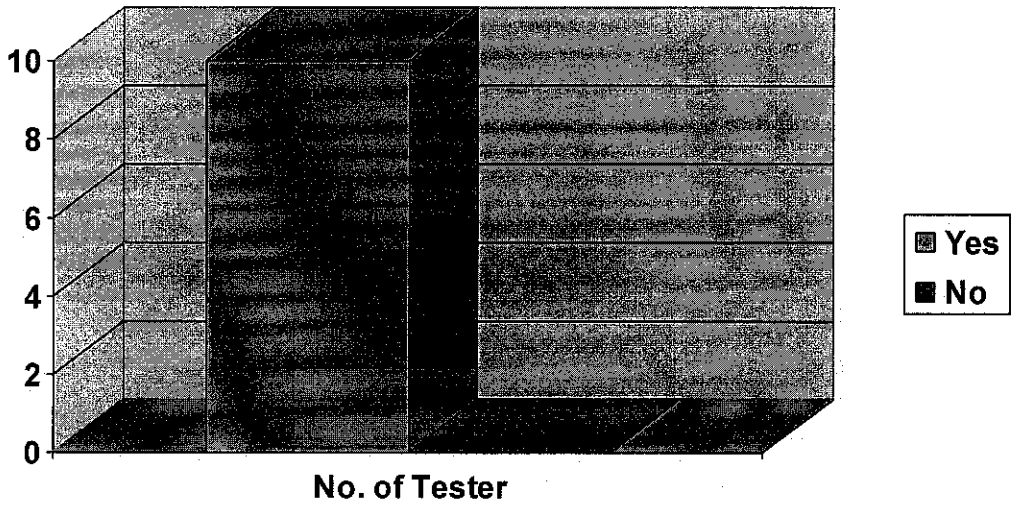


Figure 4.4: Usability Test Result 3

The result shows that the most of the user feel that they can type all the codes better in the program with the guide from the system features. This condition can lead to the objective which is to reduce typing error in writing PHP codes.

4.2 Result for Product Comparison

	Vphp	Frontpage 2003	Dreamweaver	Editplus 2.0
Codelist	**Yes**	**No (HTML only)**	**No (HTML only)**	**No**
Autocomplete	**yes**	**No**	**No**	**No**
Codetips	**yes**	**No**	**No**	**No**
Coloured syntax	**yes**	**Yes**	**Yes**	**Yes**
Visual guide	**yes**	**yes**	**yes**	**no**
	5/5	**2/5**	**2/5**	**1/5**

Figure 4.5: Product Comparison Result

For product comparison, the task is to compare the product features with other similar product. The comparison is not done thoroughly, it only focus on the product features rather than on the accuracy and efficiency between the programs. The thorough study cannot be done because it is very time consuming and require specialize knowledge to perform such activity. Therefore the comparison shows that there several attribute or features of Vphp that is not in other code editor program. These extra features are hoped to bring the Vphp to achieve all the objectives of the project.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

People are looking for perfection in what ever they do. Therefore, here comes another program that is hoped to be the perfect application for the PHP programmers. As for conclusion, the project is hoped to achieve all the object of the project. Besides that, it is also hoped that the project will benefits the entire PHP programmers. The program is also should be able to cater all the problems listed in the problems statement section.

5.2 Recommendation

For future enhancement it is recommended that this project will add much more intelligent agent in providing the editor. Other than that, the future improvement on the project is that on the visual side. It is recommended that to create another control to handle object and the coding site because the available one is not that flexible.

Reference:

1. Trevor Fenner, George Loizou, Keith Mannock and Marie-Helene Ng Cheong Vee. A Simple Interactive Development Environment for C#. Retrieved from World Wide Web: http://www.dcs.bbk.ac.uk/%7Ekeith/research/sIDE/paper/jicc7.pdf

2. Gordon Kurtenbach, George Fitzmaurice, Thomas Baudel, and Bill Buxton. The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency. Retrieved from World Wide Web:

3. http://thomas.baudel.name/Morphologie/p35-kurtenbach.pdf

4. http://www.tiobe.com/tpci.htm - PHP position as famous language

5. http://www.php.net/usage.php - PHP usage stat

6. Tim Menzies, February 20, 1998, *Evaluation Issues for Visual Programming Languages,* Artificial Intelligence Department School of Computer Science and Engineering, the University of NSW

7. T. R. G. Green, M. Petre, January 1996
<http://www.cs.dal.ca/~pcox/csci6304/References/green.pdf>