CHAPTER FIVE

CONCLUSION AND RECOMMENDATION

In summarizing the work set forth in this dissertation, it is useful to analyze the original aimed of the research. As presented in Chapter 1, the goal of this research is to implement a client/server based code inspection tool that ensures each line of codes written complied with C syntax and generates the inspection outcomes at the end of the code inspection process.

CodeIns is a prototype model. The objectives stated in the introduction have been achieved. The objectives are:

a.   To develop a CASE tool for code inspection that can detect certain types of syntax errors hidden in a program written in C programming language.

b.   To generate the inspection results in listing format.

However, due to project boundaries, there are some limitations in *CodeIns*.

5.1   System Strength

Below are the strengths of *CodeIns*.

5.1.1   Online Code Inspection

CodeIns is created online so that geographically distributed inspection participants 'meet' with people in other places through workstation at their own desk. Maintaining inspection materials online reduces paperwork, because most materials are available to the participants.

5.1.2   Cost Effective

An inspection process normally involves four to six people. Each inspection covers only a small portion of the product, and thus, a number of meetings have to be conducted to completely inspect a program of above hundred lines of source code. *CodeIns* in this respect is cost effective as it helps the participants to uncover faults before they propagate to the next phase of the life cycle.

5.1.3   User ID and Password

By giving authorize user a User ID and password, unauthorized users are prohibited from accessing any document. This is to ensure that the system is secure from any unauthorized users.

5.1.4 Simple and User-friendly Interface

The user interface in *CodeIns* is friendly and easy to understand. In addition, the web pages are designed to suit a wide spectrum of users. *CodeIns* is easy to use not only for novice users, but also for users who are unfamiliar with Web-based technology.

5.1.5 Help Module

CodeIns is included with an online Help file. The help module in *CodeIns* is available with just a click of the mouse and a selected topic would be displayed.

5.2 System limitations

System limitations exist in *CodeIns* due to some internal and external problems. Some of the limitations are browser, and User ID and password limitations.

5.2.1 Browser limitations

At present, *CodeIns* requires a VBScript support browser for execution. User using browser that does not support these features will not be able to execute the available functions in the system.

5.2.2 ID and password limitations

CodeIns should provide the database implementation to store the ID and password. It does not allow the inspector to register before using the tool.

5.3 Problems encountered

Below are problem encountered during the development of *CodeIns*.

5.3.1 Lack of Experience In Web-based programming

Due to the extensive emerging of web programming languages, it is rather difficult to master and catch up in a very limited time. Lacking in exposure resulted in limited system functions. However, learning a web programming language is fun.

5.3.2 Time consuming

Learning two major things such as web programming, web database access and the network environment for *CodeIns* are very time consuming especially for a beginner who is unfamiliar with the implementation of online web-based system. A lot of studies have to be done. Nevertheless, the knowledge gained from this project has given the impact for self-improvement and experience in information technologies.

5.3.3   PC and Network break down

During the implementation of the system, the most devasting problem encountered was the frequent breakdown between the PC and the network. This led to the waste of time to troubleshoot the problem.

5.4   Suggestions and Future Enhancements

Future developments are aimed to enhance certain functionality, either by improving the techniques applied to the existing functions or by extending it with new functionality.

To improve the functionality of a system means to enhance the mechanisms supporting the operation of such system. In the case of the developments presented on this work, such enhancements can be achieved by adding and refining methods to increase the system performance and usability such as presented below. There are several enhancements that could extend the usability of the developed system.

5.4.1   Interactive and Context Sensitive Help

Currently, *CodeIns* only provides simple help module. It should be interactive and context sensitive help so that when a user seeking for help can access the relevant information quickly.

5.4.2   Support Various Popular Browser

As stated earlier, *CodeIns* requires Internet Explorer for execution. In future, *CodeIns* can be enhanced to fulfill other browser requirements such as Netscape Navigator for execution. This is because Netscape has a sizeable share in the browser market besides Internet Explorer.

5.4.3   Support of any Inspection Process

CodeIns only allows a certain process included in inspection. In future, this tool should not be tied to a particular inspection method. Instead, the tool should be flexible in its enforcement of the inspection process, in order to tailor to the process it enforces.

5.5   Conclusion

Although the application is only a prototyped model, it can be enhanced to be a fully working application that can be used by the inspectors.

The long-term objective of this work is to develop a more flexible support tool for the inspection team which includes all the *moderator, author* and *inspector*. The leading step towards good implementation is based on a thorough understanding of the tools to use. In the case of this research, its first part was devoted to understand the features of the tools, which is the web based language, and the context of its use, which is the client/server system.

There are still many areas for improvement *in CodeIns*, in terms of implementing a comprehensive code inspection tool. With the first step taken, enhancements could still be made with more features added for the future version.