Faculty of Computer Science and Information Technology
University of Malaya

A History of Artificial Intelligence

Tang Su-Ai (WET 010152)

Under The Supervision of
Miss Mangalam Sankupellay

Moderator
Puan Norisma Idris

Session 2003/2004

A Graduation Exercise Submitted to the
Faculty of Computer Science and Information Technology
University of Malaya
In partial fulfillment of the requirements for the Bachelor of Information Technology Degree

Abstract

The History of Artificial Intelligence (AI) Website is a dynamic web-based one stop centre that offers comprehensive information about the history of Artificial Intelligence. It provides an efficient method of information searching and browsing with contemporary and attractive user-friendly and hypermedia-based user interface design. On the maintenance side, the information is well organized in a way that the web administrator can easily add on or modify the website effortlessly.

The primary objectives of the website are to gather all the information related to the history of AI and provide a comprehensive source as well as a centralized repository of information to store history of AI. This is to provide a complete website that the Internet users can gain and learn more on increasingly popular field of AI.

Functions that are provided by the website include searching capability that allows users to quickly locate the information the users are looking for. Users can also browse for the history of AI according to researchers, fields or in chronological order. A recent event section is provided to place all the recent events or developments of AI. Other miscellaneous sections are introduction of AI, about the website, etc. All the information on the website is created in hypermedia format.

Overall, at the end of the development cycle, deploying this website will be helpful and practical in promoting the ever expanding knowledge of Artificial Intelligence to the World Wide Web users.

Acknowledgement

The History of Artificial Intelligence Website has been successful developed with supports, advices, guidance, contributions and ideas from many people.

First of all, I would like to express my deepest appreciation to my supervisor, Miss Mangalam Sankupellay. I would like to thanks her for being such a gracious supervisor, also, for her valuable guidance and advices. She is unwearied in giving me guideline and explanation about the system. Under her supervises, I manage to understand the system that I am going to build.

Beside that, I would like to raise my thanks to Puan Norisma Idris, moderator of my project for spending time to moderate and giving constructive ideas to this project. The ideas help me a lot in building a comprehensive system.

Finally, I would like to convey my special thanks to all my friends especially Tan Chia Wei for spending lot his time giving me the assistances, suggestions and helps to develop the website. They have given me a lot of valuable advice, suggestion and guidance especially in programming and system development.

Table of Content

Chapter 6 – Testing

Chapter 7 – System Evaluation

List of Figures

Chapter 1 - Introduction

Chapter 2 – Literature Review

Chapter 3 - Methodology

Chapter 4 – System Analysis and Design

Chapter 5– System Implementation

List of Tables

Chapter 3 - Methodology

Chapter 4 – System Analysis and Design

Chapter 5 – System Implementation

Abstract

The History of Artificial Intelligence (AI) Website is a dynamic web-based one stop centre that offers comprehensive information about the history of Artificial Intelligence. It provides an efficient method of information searching and browsing with contemporary and attractive user-friendly and hypermedia-based user interface design. On the maintenance side, the information is well organized in a way that the web administrator can easily add on or modify the website effortlessly.

The primary objectives of the website are to gather all the information related to the history of AI and provide a comprehensive source as well as a centralized repository of information to store history of AI. This is to provide a complete website that the Internet users can gain and learn more on increasingly popular field of AI.

Functions that are provided by the website include searching capability that allows users to quickly locate the information the users are looking for. Users can also browse for the history of AI according to researchers, fields or in chronological order. A recent event section is provided to place all the recent events or developments of AI. Other miscellaneous sections are introduction of AI, about the website, etc. All the information on the website is created in hypermedia format.

Overall, at the end of the development cycle, deploying this website will be helpful and practical in promoting the ever expanding knowledge of Artificial Intelligence to the World Wide Web users.

Acknowledgement

The History of Artificial Intelligence Website has been successful developed with supports, advices, guidance, contributions and ideas from many people.

First of all, I would like to express my deepest appreciation to my supervisor, Miss Mangalam Sankupellay. I would like to thanks her for being such a gracious supervisor, also, for her valuable guidance and advices. She is unwearied in giving me guideline and explanation about the system. Under her supervises, I manage to understand the system that I am going to build.

Beside that, I would like to raise my thanks to Puan Norisma Idris, moderator of my project for spending time to moderate and giving constructive ideas to this project. The ideas help me a lot in building a comprehensive system.

Finally, I would like to convey my special thanks to all my friends especially Tan Chia Wei for spending lot his time giving me the assistances, suggestions and helps to develop the website. They have given me a lot of valuable advice, suggestion and guidance especially in programming and system development.

Chapter 1 – Introduction

1.1 Introduction

Artificial Intelligence (AI) is not a new term nowadays. It is a common vocabulary for most of the people today especially for those who is interested in Computer Science. The term Artificial Intelligence was first coined by John McCarthy in 1956 when he proposed that "intelligence can in principle be so precisely described that a machine can be made to simulate it" at the Dartmouth conference. He now defines A.I. as "the science and engineering of making intelligent machines, especially intelligent computer programs". AI has a rapid development and there are many fields in AI today. Some of the popular ones are neural networks, natural language processing, expert system, robotics, game playing and computer vision. Although we can find a lot of information about AI, there is still a lack of information about the history of AI. The resources available are not comprehensive and some of the resources online are not user friendly. Beside that, the dull and uninteresting representation of the information may make the readers feel bored with the reading materials

1.2 Project Overview

This project is aimed at building a dynamic website about the history of artificial intelligence. The website is going to provide the users with comprehensive information about the history of AI. Through this website, readers can have a clearer picture on the development of AI, the fields or topics of AI, as well as the prominent people in the

world of AI. In addition to that, the website will be user friendly with the use of hypermedia where the readers can navigate around to search for the information they need. Beside that, the developed website will be interesting enough so that the users would not feel bored as they browse through the history of AI.

1.3 Project Objective and Goals

1.3.1 Goals

- Develop a dynamic website on the history of artificial intelligence.
- Develop a hypermedia and user friendly website.
- Develop a website with comprehensive information of history of AI.

1.3.2 Objectives

- Provide a comprehensive source on the history of AI as well as the current research on AI for those who are interesting in this field.
- Enable the webmaster to update the website easily.
- To create a centralized repository of information to store history of AI.
- Provide the Internet users with an interesting website where they can gain a lot of knowledge.

1.4 Project Scope

The website developed is targeted to all those who are interested to learn about AI especially on the history of AI. Compared to those typical websites where the contents

are static, this website stores all its contents in a centralized repository for easier maintenance as well as more flexibility in producing its web pages. Furthermore, it will be a hypermedia-based system that is appealing for the Internet users to browse on without feeling bored.

1.5 Project Expectation

The expectation of this project is to build a dynamic website that is interesting and full of comprehensive information about the history of AI. The website should allow easy navigation by the users and at the same time provide ease of maintenance for the webmaster.

1.6 Project Schedule

The project schedule will be divided into two phases, phase I and phase II. Phase I is about the description of the system which includes the literature review, analysis and design. Activities carried out in this phase are preliminary investigation, literature review, system analysis and system design. Phase II is about the development of the system. Activities which are going to carry out in this phase are coding, evaluation and testing and implementation.

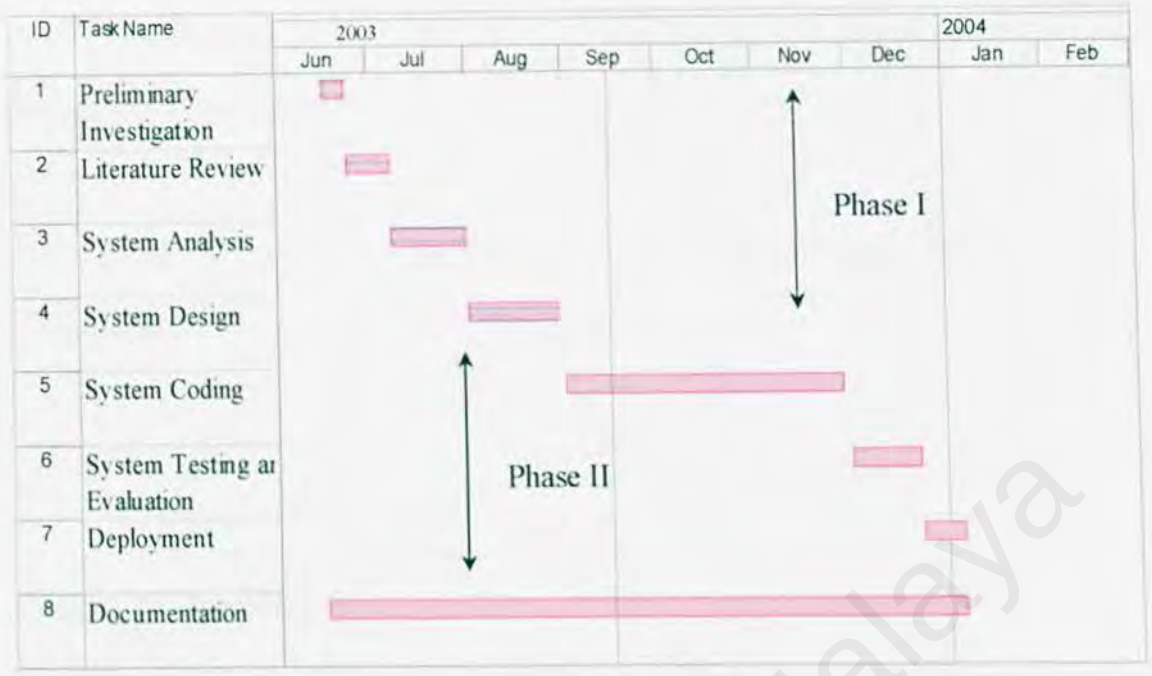

Figure 1.1: Project Schedule of the Website

1.6.1 Schedule in Phase I

Preliminary Investigation is an activity to investigate the system that is going to be built and this activity needs around one week's time. The other important activity is Literature Review which is to perform a background study on the system and it requires about two weeks. Another three weeks are used to complete the system analysis based on the most effective methodology. System Design, which includes program design, database design and interface design, needs around four weeks time to be carried out.

1.6.2 Schedule in Phase II

Coding is a complicated activity and it needs quite a long time, a total of twelve weeks, to be completed. Coding for the system is then followed by system evaluation and testing which is aimed at finding out the errors and bugs for the developed system. This

activity needs three weeks' time. At the end, another two weeks is needed for deployment of the website.

The only activity, which will be done all the time, during Phase I and Phase II, is documentation.

1.7 Chapters Summary

Chapter 1 – Introduction

This chapter will describe the overview of the project. The sections included in this chapter are project overview, project objectives and goals, project scope, project expectation and project schedule.

Chapter 2 – Literature Review

This chapter elaborates on the background study of this project. The purpose of the study is to get the all information needed to carry out the project. This chapter includes finding and analyzing the existing project as well as extensive research on the available programming technology, database, etc. With this, the best method to develop the system can then be chosen.

Chapter 3 – Methodology

This chapter will describe the methodology of the system. The software model will include in this chapter too. This chapter analyzes the system requirements, including

functional and nonfunctional requirements. This chapter also includes hardware and software requirements.

Chapter 4 – System Analysis and Design

This chapter contains the user interface design, database design, data flow and program design.

Chapter 5 – System Implementation

This chapter contains the description on system implementation where the modules and algorithms are changed into instruction using the programming language.

Chapter 6 – System Testing

This chapter is about the description on the testing of the system. It make sure that the system can function and operate based on the requirements and specification of the system.

Chapter 7 – System Evaluation

This chapter will discuss the result, problem and solution, as well as the strength and constraints of the system. It also includes future enhancement, suggestion and conclusion on the system.

Chapter 2 – Literature Review

2.1 Introduction

Literature review is the background study to obtain the knowledge and information needed in this project. The process includes finding and analyzing the existing systems that are similar to this system, and then reviewing and comparing their weakness and strength. This comparison is essential to get a clearer picture of the project that is going to be carried out. Beside that, these research and comparison will also allow us to capture the requirements of the system. The extensive research on the programming languages, databases, web technology, etc. in this section is needed to decide on the best method to develop the system.

2.2 What Is Artificial Intelligence

The term Artificial Intelligence was first coined by John McCarthy in 1956 when he proposed that "intelligence can in principle be so precisely described that a machine can be made to simulate it" at the Dartmouth conference. He now defines A.I. as "the science and engineering of making intelligent machines, especially intelligent computer programs". According to the American Association For Artificial Intelligence, artificial intelligence is defined as "the scientific understanding of the mechanisms underlying thought and intelligent behavior and their embodiment in machines". Anyway, there are many other definitions of artificial intelligence, such as "It is the science and engineering of making intelligent machines, especially intelligent computer programs. It

is related to the similar task of using computers to understand human intelligence, but AI does not have to confine itself to methods that are biologically observable", and "Artificial Intelligence (AI) is concerned with intelligent behavior in artifacts. Intelligent behavior involves perception, reasoning, learning, communicating and acting in complex environments". AI is generally associated with Computer Science, but it also has many important links with other fields such as Psychology, Cognition, Biology, Math and Philosophy, among many others (AI Depot, 2002). There are many fields in AI, some of the more popular ones are neural networks, natural language processing, expert system, robotics, game playing and computer vision.

2.2.1 Neural Networks

AI was first developed as an outgrowth of the study of the human brain and nervous system. Neural networks have great closeness to the structure of human brain. The brain and nervous system are composed of cells called neurons where as neural networks are made up of interconnected processing elements which called units (Top-Down Market Research, 2003). The network consists of several "layers" of neurons which are an input layer, hidden layers, and output layers. Input layer takes the input and distribute it to the hidden layers (so-called *hidden* because the user cannot see the inputs or outputs for those layers). These hidden layers do all the necessary computation and output the results to the output layer, which outputs the data to the user (Matthews, 2003).

The first neural network computing model was introduced by Warren McCulloch, a neurophysiologist and Walter Pitts, a young mathematician in 1943. They wrote a paper

on how neurons might work and modeled a simple neural network with electrical circuit. Recent work on neural network includes Boltzmann machines, Hopfield nets, competitive learning models, multilayer networks, and adaptive resonance theory models.

2.2.2 Natural Language Processing (NLP)

Natural languages is the one that human use to talk to one another such as Chinese, Danish, and English. Natural language processing techniques make it practical to develop programs that make queries to database, extract information from texts, retrieve relevant documents from a collection, translate one language to another, or recognize spoken words. The successful systems that put natural language to practical use share two properties. First, they are focused on a particular domain rather than allowing discussion of any topic, and second, they are focus on a particular task rather than attempting to understand language completely (Russell & Norvig, 1995).

The first natural language processing application is a dictionary look-up system developed at Birkbeck College, London. Some of the successful NLP systems are Shrdlu, Lunar, TAUM-METEO and SYSTRAN.

2.2.3 Expert Systems

Expert systems, or known as "knowledge-based systems" is the design / development of system that could match (or in some cases, exceed) the performance of human experts

on narrowly defined tasks (Russell & Norvig, 1995). Expert systems are focused on knowledge representation.

The first expert system is DENDRAL interpreted the output of a mass spectrometer, a type of instrument used for analysis if the structure of organic chemical compounds. An early statement of the DENDRAL philosophical perspective can be found in Feigenbaum, Buchanan, and Lederberg (1971).

2.2.4 Robotics

The Robot Institute of America defines a robot as a programmable, multifunction manipulator designed to move material, parts, tools or specific devices through variable programmed motions for the performance of a variety of tasks. Reichardt (1978) survey the history and future of robots and grouping them into four categories. First, strictly mythological, fictional, or fraudulent; second, working, but nonelectronic; third, controlled by very special-purpose electronic or electromechanical hardware; fourth, controlled by general-purpose computers (Russell & Norvig, 1995).

George Engelberger and George Devol developed the first useful industrial robots in the late of 1950s. Engelberger is also known as "father of robotics". The first modern mobile robot was the "Hopkins Beast", built in the early 1960s and the first general-purpose robot was "Shakey".

2.2.5 Game Playing

Games are ideal domains for exploring the capabilities of computational intelligence. The rules are fixed, the scope of the problem is constrained, and the interactions of the players are well defined. A game can be defined by the initial state, the operators, a terminal test and a utility or payoff function (Russell & Norvig, 1995).

The first functioning (and nonfraudulent) game playing machine was designed and biuilt by Spanish engineer, Leonardo Torrès y Quevedo around 1890. Alan Turing wrote the first actual computer program capable of playing a full game of chess in 1951 but the program never actually ran on a computer.

2.2.6 Computer Vision

Computer vision, also referred to as machine vision is to make computers understand and interpret visual information. The field of computer vision has histories of at least two centuries. The first is the history of mechanical devices designed to aid human perception, which aims to replace human sight altogether. The second is the history of automata, whose construction was especially popular in the seventeenth and eighteenth centuries, which imitated human's or animal's bodily functions (Nalwa, 1993).

It was Leonardo Torres in 1913 who expressed a fundamentally new idea of using automata for productive labor by replacing a man by a machine; machines that would "accomplish the results which a living person obtains"

2.3 Introduction To History Of Artificial Intelligence

History can be defined as what has happened at the past. Through the history of artificial intelligence (AI), we can know what has happened in this field, how is the development of artificial intelligence as well as who's who in the world of AI. With this knowledge, we can further comprehend the whole idea of AI as well as foresee the future of AI. The intellectual roots of artificial intelligence and the concept of intelligent machines may be found in Greek mythology. Greek myths of Hephaestus and Pygmalion have incorporated the idea of intelligent robots. Anyway, the term "Artificial Intelligence" was first coined by John McCarthy in a proposal for the Dartmouth conference in 1955. Nowadays, AI has a rapid development where we can find the robot pets in the market and the robot with a face that expresses emotions.

2.4 Research on Existing Website

There is no system / websites which is focused wholly on the history of Artificial Intelligence. For the websites that have been found, history of AI is just part of the contents of the websites. Among them, there is only a websites which provides a more detail information on history of AI. For the others, history of AI is just briefly explained and the information given is not comprehensive.

2.4.1 Brief History Of Artificial Intelligence

On this website, history of AI is just a part of its contents. Anyway, the history of AI in this websites is quite comprehensive. The history is stated according to year and the system is hypermedia where the users can navigate through who's who in the world of

AI as well as the development and finding inside AI. This website is built by American Association of Artificial Intelligence. It includes many sections related to AI such as applications of AI, AI in news, AI overview, AI industry statistic as well as links to other related sites.

Strength:

- The websites is colorful where the readers would not feel bored when reading through it.
- Contains cartoons and drawing that make reading on AI fun, exciting and interesting.
- Information provided in this website is quite comprehensive as it covered many topics about AI.
- The information on history of AI is quite comprehensive and it is in hypermedia format.

Weakness:

- The website is lacked of animation and flash.
- It is lack of arrangement and the information provided is unorganized

Interface:

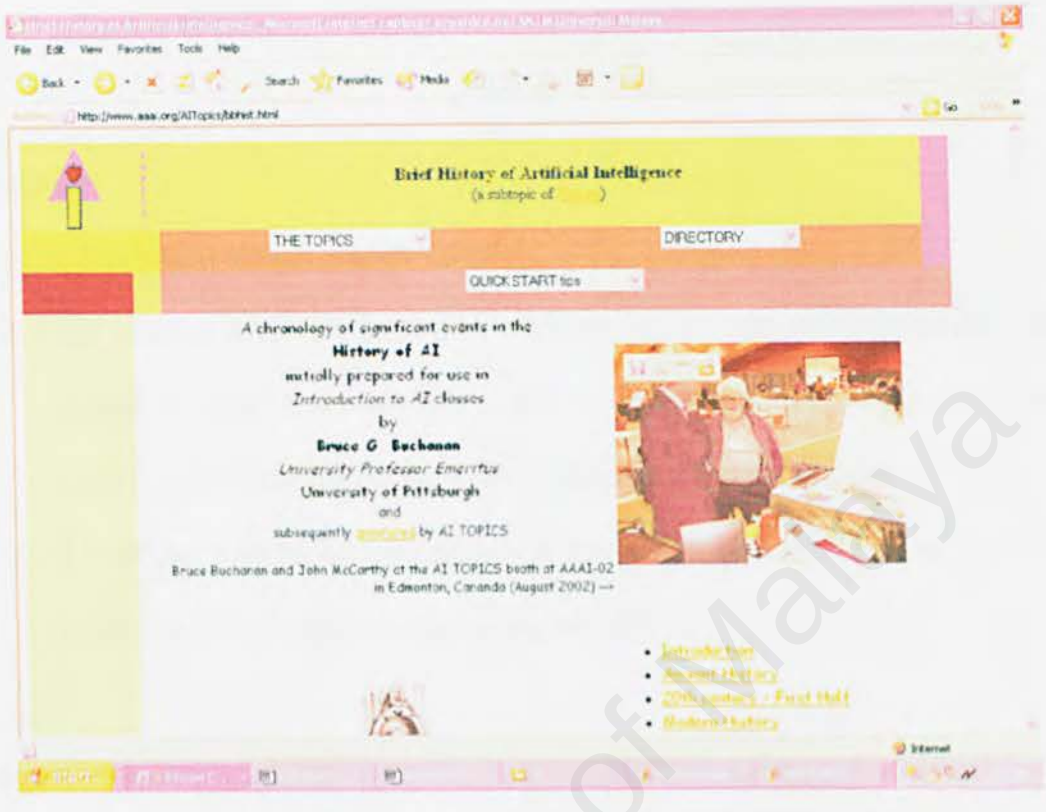

Figure 2.1: Interface of Brief History of Artificial Intelligence

2.4.2 Stottler Henke – Smarter Software Solutions

This is a website for a company. Just like many others websites, the history of AI is just a part of the contents of this website. The information available in this website is more on the company itself. Anyway, it also provides quite a lot of information about the history of AI.

Strength:

- It provides quite a lot of information about the history of AI.

- The information of the history of AI is quite systematic and it is arranged according to years.

Weakness:

- It is not a hypermedia-based system.
- The information about the history of AI maybe still not comprehensive enough as readers cannot find more information about the prominent people in the world of AI as well as the explanations about some of the terms or fields.
- It is neither colorful nor interesting – it does not contain animation or flash. As a result, the website looked quite boring and dull.

Interface:

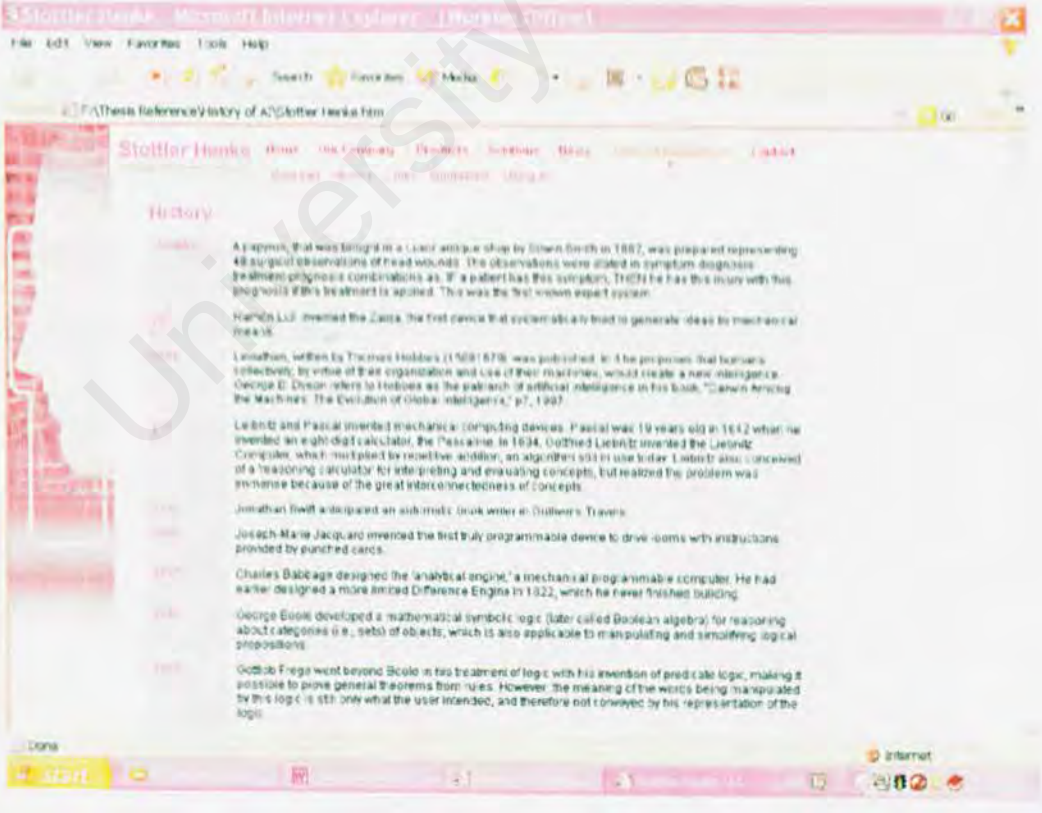

Figure 2.2: Interface of Stottler Henke – Smarter Software Solutions

2.4.3 Computer History Museum

As its name implies, the content of this website revolved around the history of computer. Majority of the contents discuss about history of computer on the whole, with the history of AI is just part of them. Although information about the history of AI is not comprehensive, they are well organized and well presented together with photos and distinctive color scheme.

Strength:

- Well organized information
- Well presented – colors are cleverly used to contrast the contents.
- A lot of related photos are shown, allowing the readers to have more understanding on the topic presented.

Weakness:

- Not comprehensive
- Not in hypermedia format
- No animation or flash.

Interface:

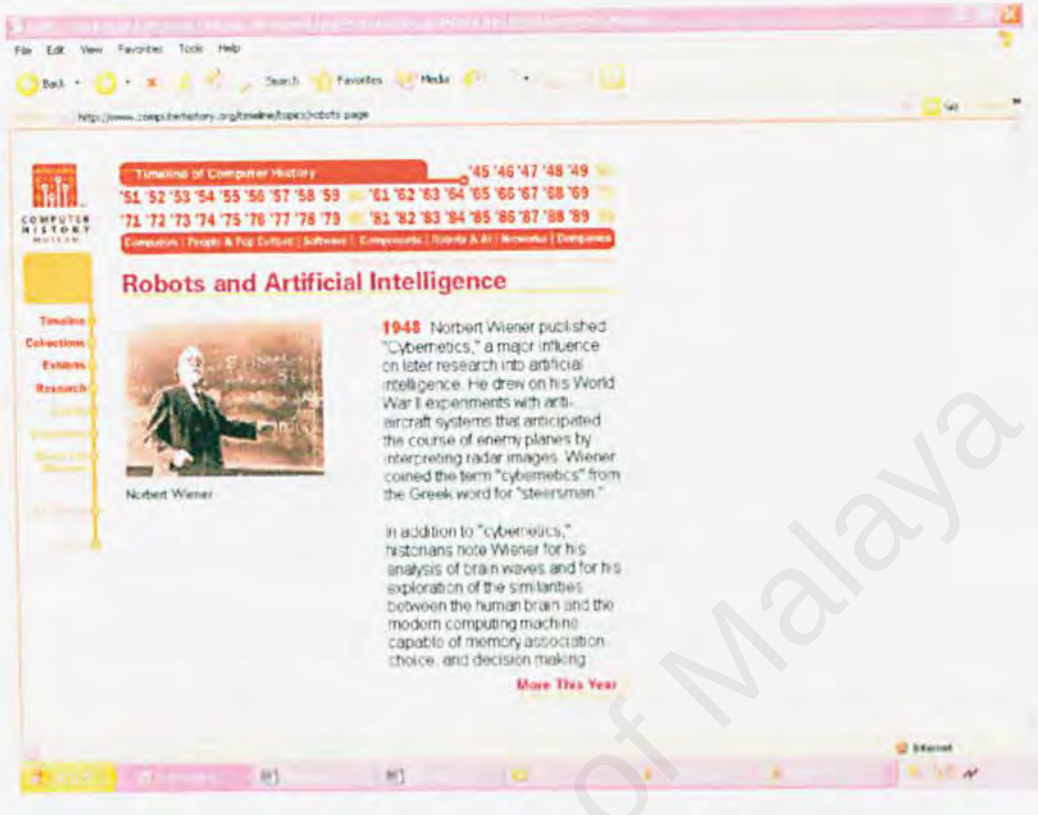

Figure 2.3: Interface of Computer History Museum

2.5 What Is Dynamic Website

Dynamic website is a website that dynamically produces web pages based on the interests and the decisions of the viewer. Therefore, a visitor can view web pages which are unique to their interests, and information can be incorporated from divergent files within a database. Dynamic website has much more strength over the static website. Static website is a website where the content consists of some HTML code that was typed directly into a text editor and saved as an .htm or .html file. Thus, the author of the page has already completely determined the exact content of the page in HTML, at some time before any user visits the page or website (Gdalevich, 2002). One of the key

strengths of dynamic website in this project context would be its ability to change or enhance the contents of the website rapidly compared to those "hard-coded" static websites, where a minor change often requires renovation of the whole websites. In other words, dynamic website promotes extensibility, maintainability as well as flexibility.

Here are some of the strengths of dynamic website:

- Creating value. It allows users to have access to information and allowing them to interact with that information, provides value.
- Less costly. Depending on the site, it can be less costly to create a dynamic rather than static site. Dynamic sites tend to have less actual pages but appear to have thousands of pages. A dynamic website will be cheaper than a static website if there are a lot of repetitive pages. Beside that, the maintenance costs for dynamic website are also lower since changes made in the existing information system automatically get reflected on the web site. Once the website has been set up, no HTML experience is needed to maintain the data in that site.
- Centralizing data. Dynamic web sites involve the use of database technology where large amounts of data can be stored and organized in an RDBMS. A centralized data depository can reduce duplication and improve data accuracy.
- Customizing content. Dynamic website enables the users to view the content based on their needs or preferences thus save their time. Customization is a very powerful technique that keeps a users attention and makes them feel like an individual (Cobbe, 2003).

2.6 What Is Hypermedia

Hypermedia is an acronym which combines the words hypertext and multimedia (De Bra, 2002). Hypertext is the concept of interrelating information elements using links to enable easy access of related information. Hypertext allows an author to create interrelated information and allows a reader to traverse the information through links, for example, to navigate from one page to another using these links. Hypertext systems include many navigation, annotation and structural features, which take advantage of the linked structure for the ease of readers. Well designed hypertext enables people to read and comprehend information more effectively than traditional documents. At the same time, it enables readers to access information in the order most appropriate to their purposes.

2.7 Web Technology / Architectures

The World Wide Web or the Internet is basically a complexly wired client/server technology whereby Internet surfers act as clients to access information from servers located all around the world. Thus in this section, client/server architectures are discussed and reviewed for the purpose of choosing a suitable architecture for this project. Client/server architectures discussed in the following are the more popular ones – two-tier and three-tier client/server architectures.

2.7.1 Two-tier Architecture

In this architecture, the three components of an application (i.e. presentation, processing, and data) are divided into two entities (tiers) – client application code and database server (Figure 2.4). A robust client application development language and a versatile mechanism for transmitting client requests to the server are essential for a two tier implementation. Presentation is handled exclusively by the client, processing is split between client and server, and data is stored on and accessed via the server. The PC client assumes the bulk of responsibility for application (functionality) logic with respect to the processing component, while the database engine - with its attendant integrity checks, query capabilities and central repository functions - handles data intensive tasks. In a data access topology, a data engine would process requests sent from the clients. Currently, the language used in these requests is most typically a form of SQL. Sending SQL from client to server requires a tight linkage between the two layers. To send the SQL the client must know the syntax of the server or have this translated via an API (Application Program Interface). It must also know the location of the server, how the data is organized, and how the data is named. The request may take advantage of logic stored and processed on the server which would centralize global tasks such as validation, data integrity, and security. Data returned to the client can be manipulated at the client level for further sub selection, business modeling, "what if" analysis, reporting, etc.

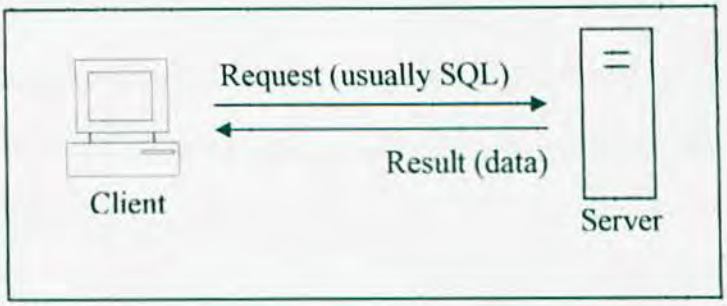

Figure 2.4 - Data Access Topology for two-tier architecture. Majority of functional logic exists at the client level

The most compelling advantage of a two-tier environment is application development speed. In most cases a two-tier system can be developed in a small fraction of the time it would take to code a comparable but less flexible legacy system. Using any one of a growing number of PC-based tools, a single developer can model data and populate a database on a remote server, paint a user interface, create a client with application logic, and include data access routines. Most two-tier tools are also extremely robust. These environments support a variety of data structures, including a number of built in procedures and functions, and insulate developers from many of the more mundane aspects of programming such as memory management. Finally these tools also lend themselves well to iterative prototyping and rapid application development (RAD) techniques, which can be used to ensure that the requirements of the users are accurately and completely met.

Since the bulk of application logic exists on the PC client, the two-tier architecture faces a number of potential version control and application re-distribution problems. A change in business rules would require a change to the client logic in each application in

a corporation's portfolio which is affected by the change. Modified clients would have to be re-distributed through the network - a potentially difficult task given the current lack of robust PC version control software and problems associated with upgrading PCs that are turned off or not "docked" to the network.

System security in the two-tier environment can be complicated since a user may require a separate password for each SQL server accessed. The proliferation of end-user query tools can also compromise database server security. The overwhelming majority of client/server applications developed today are designed without sophisticated middleware technologies which offer increased security. Instead, end-users are provided a password which gives them access to a database. In many cases this same password can be used to access the database with data-access tools available in most commercial PC spreadsheet and database packages. Using such a tool, a user may be able to access otherwise hidden fields or tables and possibly corrupt data.

Client tools and the SQL middleware used in two-tier environments are also highly proprietary and the PC tools market is extremely volatile. The client/server tools market seems to be changing at an increasingly unstable rate. In 1994, the leading client/server tool developer was purchased by a large database firm, raising concern about the manufacturer's ability to continue to work cooperatively with RDBMS vendors which compete with the parent company's products. All of this complicates implementation of two-tier systems - migration from one proprietary technology to another would require a firm to scrap much of its investment in application code since none of this code is portable from one tool to the next.

2.7.2 Three tier

The tree tier architecture (Figure 2.5) attempts to overcome some of the limitations of the two-tier scheme by separating presentation, processing, and data into separate, distinct entities (tiers). The same types of tools can be used for presentation as were used in a two-tier environment, however these tools are now dedicated to handling just the presentation. When calculations or data access is required by the presentation client, a call is made to a middle tier functionality server. This tier can perform calculations or can make requests as a client to additional servers. The middle tier servers are typically coded in a highly-portable, non-proprietary language such as C. Middle-tier functionality servers may be multi-threaded and can be accessed by multiple clients, even those from separate applications.

Although three-tier systems can be implemented using a variety of technologies, the calling mechanism from client to server in such as system is most typically the remote procedure call or RPC. Since the bulk of two-tier implementations involve SQL messaging and most three-tier systems utilize RPCs, it is reasonable to examine the merits of these respective request/response mechanisms in a discussion of architectures. RPC calls from presentation client to middle-tier server provide greater overall system flexibility than the SQL calls made by clients in the two-tier architecture. This is because in an RPC, the requesting client simply passes parameters needed for the request and specifies a data structure to accept returned values (if any). Unlike most two-tier implementations, the three tier presentation client is not required to "speak" SQL. As such, the organization, names, or even the overall structure of the back-end

data can be changed without requiring changes to PC-based presentation clients. Since SQL is no longer required, data can be organized hierarchically, relationally, or in object format. This added flexibility can allow a firm to access legacy data and simplifies the introduction of new database technologies.

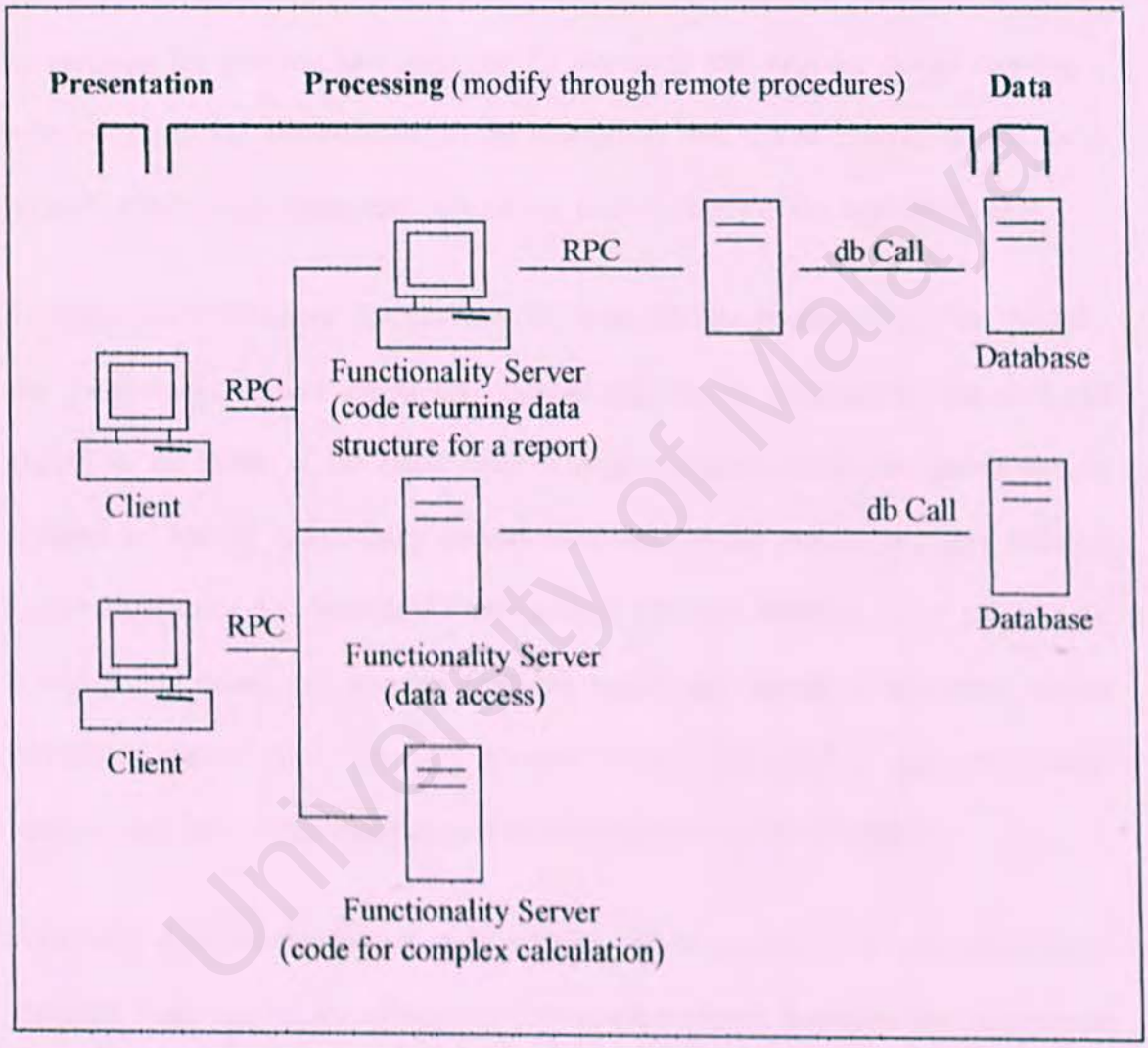

Figure 2.5 - Three Tier Architecture. Most of the logic processing is handled by functionality servers. Middle-tier code can be accessed and utilized by multiple clients

In addition to the openness stated above, several other advantages are presented by this architecture. Having separate software entities can allow for the parallel development of individual tiers by application specialists. It should be noted that the skill sets required to develop client/server applications differ significantly from those needed to develop mainframe-based character systems. As examples, user interface creation requires an appreciation for platform and corporate UI standards and database design requires a commitment to and understanding of the enterprise's data model. Having experts focus on each of these three layers can increase the overall quality of the final application.

The three tier architecture also provides for more flexible resource allocation. Middle-tier functionality servers are highly portable and can be dynamically allocated and shifted as the needs of the organization change. Network traffic can potentially be reduced by having functionality servers strip data to the precise structure required before distributing it to individual clients at the LAN level. Multiple server requests and complex data access can emanate from the middle tier instead of the client, further decreasing traffic. Also, since PC clients are now dedicated to just presentation, memory and disk storage requirements for PCs will potentially be reduced.

Modularly designed middle tier code modules can be re-used by several applications. Reusable logic can reduce subsequent development efforts, minimize the maintenance work load, and decrease migration costs when switching client applications. In addition, implementation platforms for three tier systems such as OSF/DCE offer a variety of additional features to support distributed application development. These include integrated security, directory and naming services, server monitoring and boot

capabilities for supporting dynamic fault-tolerance, and distributed time management for synchronizing systems across networks and separate time zones.

There are of course drawbacks associated with three tier architecture. Current tools are relatively immature and require more complex 3GLs for middle tier server generation. Many tools have under-developed facilities for maintaining server libraries - a potential obstacle for simplifying maintenance and promoting code re-use. More code in more places also increases the likelihood that a system failure will effect an application so detailed planning with an emphasis on the reduction/elimination of critical-paths is essential. Three tier brings with it an increased need for network traffic management, server load balancing, and fault tolerance.

2.7.3 Comparing Two And Three Tier Development Efforts

Time to deployment is forecast in overall systems delivery time, not man hours. According to a Deloitte & Touche study, rapid application development time is cited as one of the primary reasons firms chose to migrate to a client/server architecture. Thus, strategic planning and platform decisions require an understanding how development time relates to architecture.

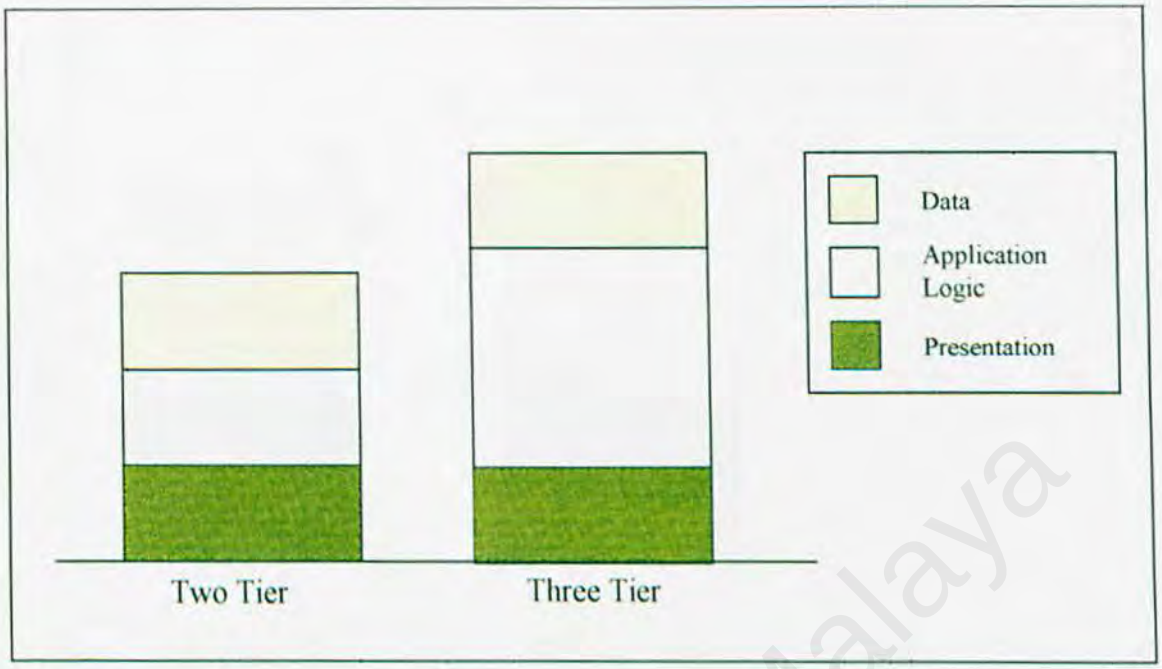

Figure 2.6 - Initial Development Effort

Figure 2.6 shows the initial development effort forecast to create comparable distributed applications using the common two tier and three tier approaches discussed above. The three tier application takes much longer to develop - this is due primarily to the complexity involved in coding the bulk of the application logic in a lower-level 3GL such as C and the difficulties associated with coordinating multiple independent software modules on disparate platforms. In contrast, the two-tier scheme allows the bulk of the application logic to be developed in a higher-level language within the same tool used to create the user interface.

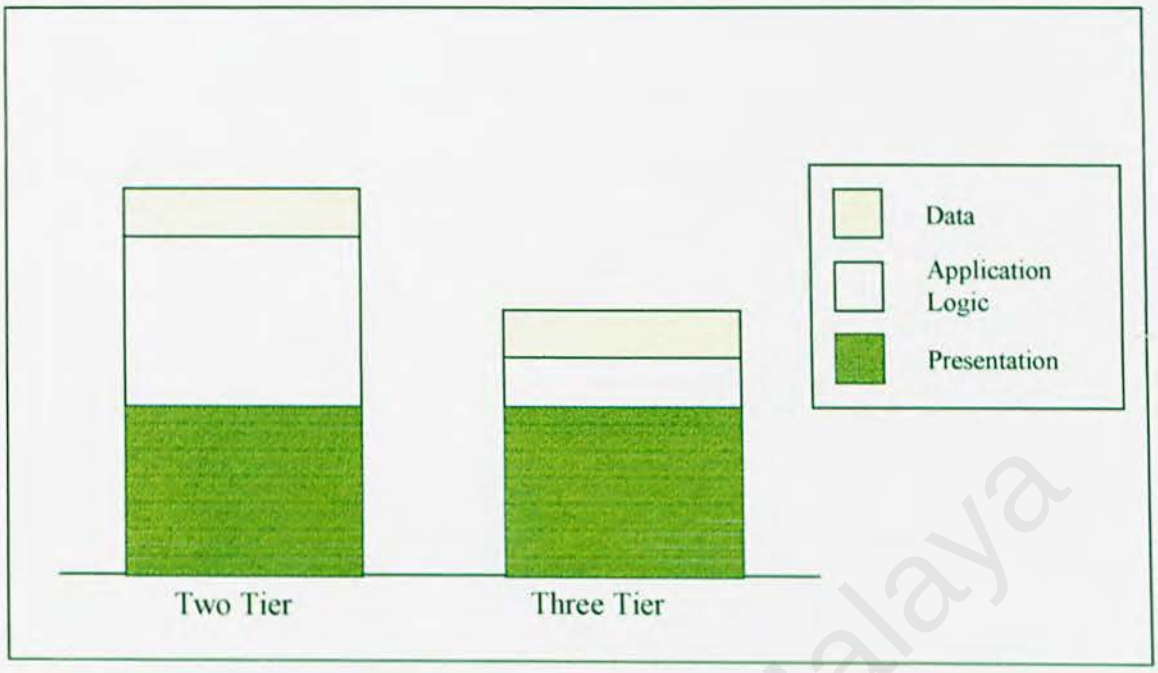

Figure 2.7 - Subsequent Development Efforts

Subsequent development efforts may see three-tier applications deployed with greater speed than two tier systems (Figure 2.7). This is entirely due to the amount of middle-tier code which can be re-used from previous applications. The speed advantage favoring the three-tier architecture will only result if the three-tier application is able to use a sizable portion of existing logic. Re-use is high if data-access code can be written once and re-used whenever similar access needs arise across multiple applications. The degree of development time reduction on subsequent efforts will grow as more client/server applications are developed with a significant library of re-usable, middle-tier application logic.

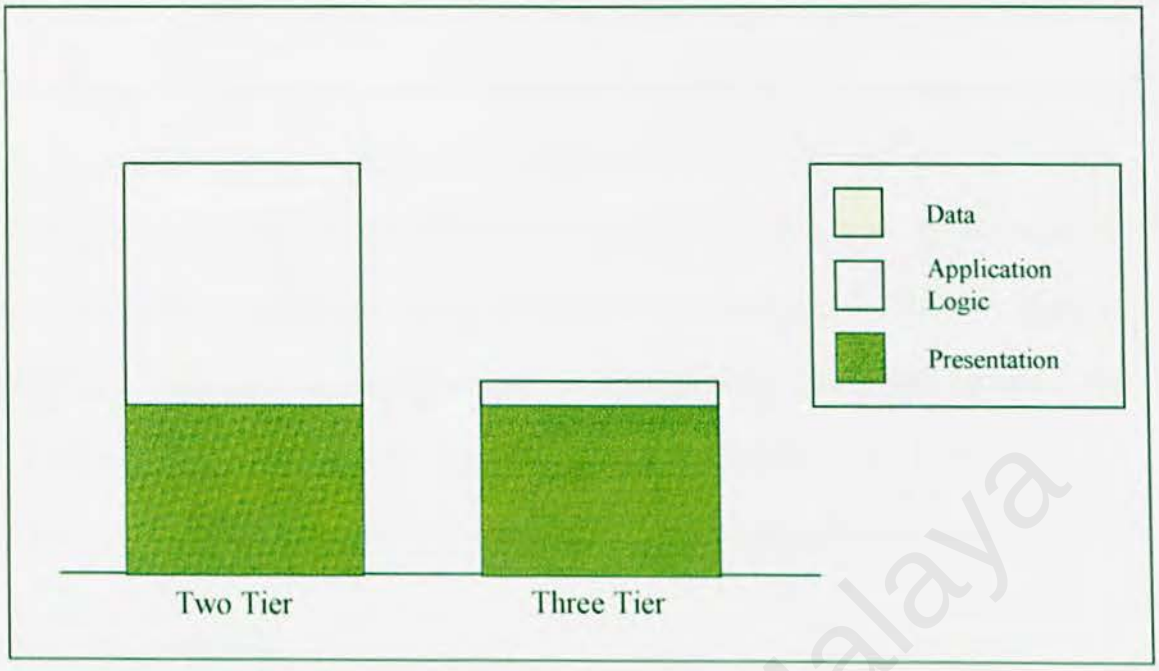

Figure 2.8 - Client Tool Migration

Figure 2.8 makes the important case for code savings when migrating from one client development tool to another. It was stated earlier that client tools are highly proprietary and code is not portable between the major vendor packages. The point was also made that the PC tools market is highly volatile with vendor shake outs and technical "leapfrogging" common place. In a two-tier environment, moving from one PC-based client development platform to another will have to scrap their previous investment in application logic since most of this logic is written in the language of the proprietary tool. In the three-tier environment this logic is written in a re-usable middle tier, thus when migrating to the new tool, the developer simply has to create the presentation and add RPC calls to the functionality layer.

Flexibility in re-using existing middle-tier code can also assist organizations developing applications for various PC client operating system platforms. Until recently there were very few cross-platform client tool development environments and most of today's cross-platform solutions are not considered "best-of-breed". In a three-tier environment the middle tier functionality layer can be accessed by separate client tools on separate platforms. Coding application logic once in an accessible middle tier decreases the overall development time on the cross-platform solution and it provides the organization greater flexibility in choosing the best tool on any given platform.

2.8 Summary

In this section, the definition and concept of the system have been studied to avoid confusion occurs in the system architecture. The study on the existing system will help to make a clearer picture on the requirement of the system that is going to be built. At the same time, the literature review will help to choose the best tool to develop the system.

Chapter 3 – Methodology

3.1 Project Model

3.1.1 Waterfall Model

Waterfall Model is the earliest software development model. The model contains five main phases which are requirement definition, system and software design, implementation and unit testing, integration and system testing and operation and maintenance.

It presents a very high level view of what goes on during development where it is linear and sequential. After one phase of development is completed, it proceeds to another phase. Each phase will transform an input product to produce a new product as output. The new product will then become the input for the next phase.

As a result, it allows for managerial control where schedule can be set with deadline for each phase of development and a product can proceed through the development process. This can be shown at the table below:

Table 3.1: Phases and products for waterfall model

Input Product	Phase	Output Product
Communicated requirement	Requirements definition	Requirement specification documents
Requirement specification document	System and software design	Design specification document.
Design specification document	Implementation and unit testing	System Modules
System Modules	Integration and system testing	Integrated Software / Website
Integrated Software / Website	Operation and maintenance	Changed requirements.

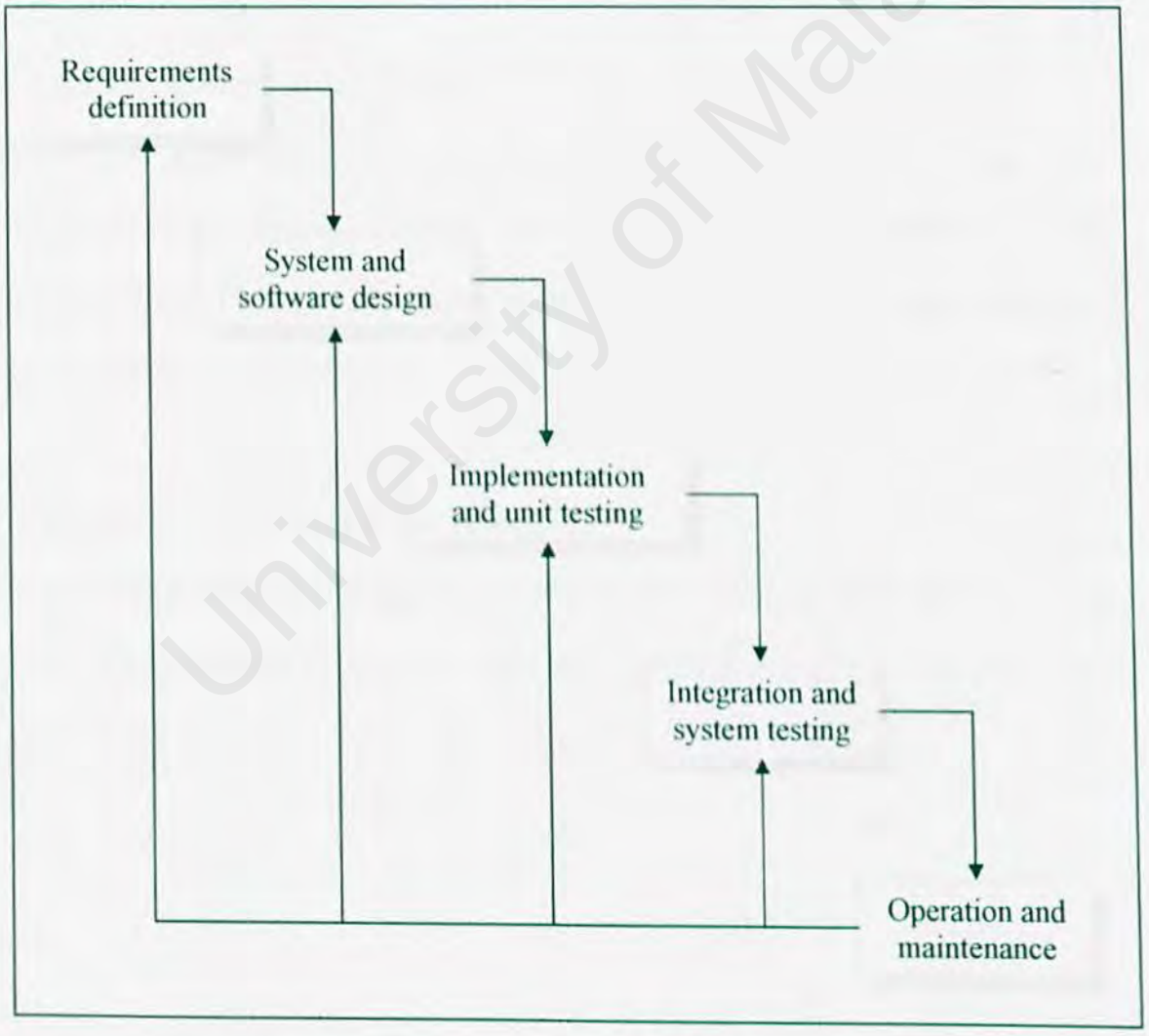

Figure 3.1: Waterfall Model

3.1.2 Activities in each phases of Waterfall Methodology

3.1.2.1 Requirement Definition

The system services or functions, requirements, constraints and goals are defined and established in this phase. It states the functions or services that should be provided by the system, the constraints or limitation of the system, the problems that may be occurred during the development as well as the goals and objectives of the established the system. The requirements are analyzed in this phase too. The delivery at the end of this phase is requirement specification document.

3.1.2.2 System and Software Design

This phase establishes an overall architecture of the system that is going to be developed. It identifies the detail on how the system will be constructed to perform necessary task. The system is established according to the requirements which are defined and analyzed in the *requirement definition* phase.

3.1.2.3 Implementation and Unit Testing

The software design is realized as a set of program or program units. These units are then tested separated to make sure that each of them conforms to the design and operates correctly.

3.1.2.4 Integration and System Testing

The individual programs or program units are integrated into one complete system. This system (website) is tested thoroughly by the developers and users to ensure the requirement are met.

3.1.2.5 Operation and Maintenance

The system is installed and put into practical use. Any further improvements or change request from the users are added/implemented to the system.

3.1.3 Rationale of Waterfall Methodology Approach

The waterfall methodology is used as software development model for this website due to the reasons stated below:-

- The methodology is simple and easy to use
- The requirements have been defined in the early state of the development and there is not much or no changes in user requirements
- Able to have presentable sub-outcome to be verified by the users at each stage
- Deadlines can be set at each stages to make sure development progress is on time
- It is suitable for the development of this website

3.2 Fact Finding Techniques

Fact-finding, also known as information gathering is the formal process to collect information about problems and requirements inside system analysis.

There are several common fact-finding techniques, they are sampling of existing documents and files, research, site visits, observation, questionnaires, interviews, discovery prototyping and joint requirements planning. Anyway, each of these techniques has its own advantages and disadvantages.

Techniques that are used to collect the information for this system are:

- Searching on the Internet
- Discussion
- Observation on available and related system.

3.2.1 Searching on Internet

A lot of valuable information can be collected through the Internet to develop this system. At the same time, by using this technique, tremendous time and cost can be saved. There are many search engines such as MSN, Yahoo and Google which can be used to search for systems that are similar to the system to be built. There are some websites which provide the history of artificial intelligence, however, much of them are not comprehensive. In order to develop a comprehensive and complete website, the contents of the website are broken down into topics and thorough search on the Internet

is conducted to obtain their related information. By reviewing the websites, clearer picture and concept on how to develop the system can be acquired.

3.2.2 Discussion

Discussions were made with the supervisor to get the information about the system to be developed. Through the discussions, the requirements of the system can be acquired.

3.2.3 Observation on available and related system

Several websites about history of artificial intelligence can be found through the Internet. Most of the websites do not provide comprehensive information. One of the website found, developed by American Association for Artificial Intelligence (AAAI), is quite similar to this system. The contents of AAAI website do not just focus on the history of AI, instead, they are diversified into different aspect of AI. It contains the brief history of artificial intelligence, researchers of artificial intelligence, news and development in artificial intelligence as well as other related topics. Beside that, there are many links in this website which linked to other AI related websites.

However, the contents of AAAI website are probably a bit uncoordinated from reader's point of view as they are not well arranged and clearly presented. Still, it provides the ideas on the system requirements.

3.3 Requirement Definition

Requirement of the system is the description of the services or functions that the system should provide as well as the constraints of the system. System requirements are generally classified into functional and non-functional requirements.

3.3.1 Functional Requirements

Functional requirements are description of the services, functions and features that should be provided by this system (website). The functional requirements should satisfy the users' need and be acceptable to the users. Functional requirements of this system (website) are as below:-

3.3.1.1 Dynamic website

The system that is developed should be a dynamic website. From the developer/administrator point of view, the website should be able to be changed, updated and enhanced rapidly at the same time the HTML codes do not need to be rewritten again.

From the users' point of view, they should be able to view the web pages based on their interests and preferences. Users should also be able to access the information that is incorporated from divergent files within a database dynamically.

3.3.1.2 Search capability

The website should provide search capability. The users should be able to search for the information they want easily using a simple search engine incorporated on the website. For simplicity, the users are limited to search according a specific field such as the researchers' name, the fields as so on instead of full-text searching. This is partly due to the limitation of the database system used which does not provide full-text searching capability.

3.3.1.3 Multimedia content

The website should provide multimedia content such as animation, pictures and possibly video and audio contents.

3.3.1.4 Hypermedia-based

The website should be hypermedia-based – it should contain the hypertext and multimedia. The users should be able to traverse the information through links and navigate from one page to another. Therefore, they are able to access the information which is most appropriate to their purposes quickly.

3.3.2 Non-functional Requirements

A non-functional requirement is a description of the features, characteristic, and attributes of the system as well as any constraints that may limit the boundaries of the proposed solution. (Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, 20001). Non-functional requirements of this system (website) are as below:-

3.3.2.1 Content rich

The contents of the website should be rich and comprehensive. It should contain as much information as possible, if not all, about the history of AI, the researchers in AI as well as the development of AI.

3.3.2.2 Fast response time

The website should have fast response time when the users are browsing it. It should also respond timely when the users are using its search feature.

3.3.2.3 Reliable

The website developed should be stable. This implies a reliable and stable web server and database system are required, together with module codes that handle all the exception cases to prevent the system from going down abruptly.

3.3.2.4 Extensible

In relation with functional requirement 3.3.1.1 – dynamic website, the website should be easily maintainable by the administrator. This includes the ease of changing the user interface as well as adding more contents to the website.

3.4 Consideration and Selection of Development Tools

In this section, development tools for the website are examined in order to select the most appropriate ones for developing the website. Development tools to be considered

are categorized into four types – web programming languages, database, data access tools and web server.

3.4.1 Consideration Of Web Programming Languages

3.4.1.1Client-Side Technologies for Dynamic Content

Each of these technologies relies on a module (or plug-in) built into the browser to process the instructions. The client-side technologies are a mixture of scripting languages, controls, and fully fledged programming languages.

3.4.1.1.1 JavaScript

JavaScript is the original browser scripting language, and is not to be confused with Java. Java is a complete application programming language in its own right, whereas JavaScript borrows some of its syntax and basic structures from Java (which in turn borrowed ideas from C). Anyway, JavaScript and Java have a different purpose and they are evolved from different origins. For example, while JavaScript can control browser behavior and content, it isn't capable in controlling features such as file handling. In fact, it is actively prevented from doing this because of security reasons.

JavaScript is much easier to learn than Java. Besides, it is designed to create small and efficient applications that can do many things, which range from performing repetitive tasks to handling events generated by the user (such as mouse clicks, keyboard responses, and so on).

3.4.1.1.2 VBScript

In Internet Explorer 3.0, Microsoft has introduced its own scripting language, VBScript, which was based on their Visual Basic programming language. VBScript was intended to be a direct competitor to JavaScript. There isn't much difference between the two (VBScript and JavaScript) in terms of functionality, but it is more a matter of personal preference. Visual Basic developers sometimes prefer VBScript because VBScript is, for the most part, a subset of Microsoft's Visual Basic language. However, there is an advantage that makes it more attractive to novice programmers, where, unlike JavaScript, it isn't case-sensitive and is therefore less fussy about the particulars of the code. Anyway, at the same time, this "advantage" has makes it a lot slower and less efficient.

The biggest drawback of VBScript is that there isn't a single non-Microsoft browser that supports it for client- side scripting. For a short while there were some proprietary plug-ins for Netscape that provided VBScript support, but these never took off. Therefore, JavaScript is much more widely used and supported. VBScript should only be considered when working on Intranet pages where it is known that all clients are IE on Windows.

3.4.1.1.3 Java applets

Java is a cross-platform language for developing applications. When it is first hit the Web in the mid- 1990s, Java created a great stir. The idea is to use Java code in the

form of applets, which are essentially Java components that can be easily inserted into web pages with the aid of the <applet> tag.

Java enjoys better functionality than scripting languages where it offers better capabilities in areas such as graphic functions and file handling. Java is able to provide these powerful features without compromising security because the applets run in what is known as a sandbox – which prevents a malicious program downloaded from the web from making damages to client system.

Microsoft and Netscape browsers both have built-in Java support via something known as the Java Virtual Machine (JVM), and there are several standard <object> and non-standard <applet> tags that are used to add Java applets to a web page. These tags tell the browser to download a Java file from a server and execute it with the Java Virtual Machine built into the browser. Of course, this extra step in the web page building phase means that Java applets can take a little while to download, and can take even longer to process once on the browser. So, while smaller Java applets (that provide features such as drop-down menus and animations) are very popular on the Web, larger ones are still not as widespread as scripted pages.

3.4.1.2 Server-Side Technologies for Dynamic Content

Each of these technologies relies on a modular attachment added onto the web server rather than the browser. Consequently, only HTML, and any client-side script, is sent

back to the browser by the web server. In other words, none of the server-side code is sent back. Server-side technologies have a more consistent look and feel than client-side ones, and it does not take that much extra learning to move between some of the server-side technologies (except CGI).

3.4.1.2.1 CGI

The Common Gateway Interface (CGI) is a mechanism for creating scripts on the server, which can then be used to create dynamic web applications. CGI is a module that is added to the web server and has been around for quite a bit longer than even ASP. Nowadays, there are a large proportion of dynamically created web pages are created using CGI and a scripting language. CGI allows the user to invoke another program (such as a Perl script) on the web server to create the dynamic web page, and the role of CGI is to pass the user supplied data to the program for processing. However, it does provide the same end result – a dynamic web application.

CGI has some disadvantages:

- It is not easy for a beginner to learn how to program such modules.
- CGI requires a lot of server resources, especially in a multi-user situation.
- It adds an extra step to the server-side model of creating dynamic content, that is, it is necessary to run a CGI program to create the dynamic page before the page is processed on the server.

Good facilities for manipulating text and communicating with other software are needed in order to use CGI. This is due to the format in which CGI receives and transmits data and the data is not easily manipulated by many programming languages. The most common programming languages that can work on any operating system for doing this are C, C++ and Perl, anyway, they are some of the more complex languages to learn. Visual Basic does not offer adequate text handling facilities, thus it is rarely used with CGI.

3.4.1.2.2 JSP

JavaServer Pages (JSP) is a technology which allows the combination of markup (HTML or XML) with Java code to dynamically generate web pages. The JSP specification is implemented by several web servers and one of its main advantages is the portability of code between different servers. JSP is very powerful and faster than ASP, and at the same time, instantly familiar to Java programmers. It allows the Java program to leverage the aspects of the Java2 platform such as JavaBeans and the Java 2 libraries. JavaServer Pages isn't directly related ASP, but it does boast the ability to embed Java code into web pages using server-side tags.

3.4.1.2.3 ASP

ASP generally relied on either of the JavaScript or VBScript scripting languages to create dynamic web page. ASP is a module (the asp.dll file) that is attached to a web server, and it then processes the JavaScript/VBScript on the web server, and turns it into HTML, before sending it into the server, rather than doing it on the browser.

ASP can use practically any of the functionality provided by Windows, such as database access, e- mailing, graphics, networking, and system functions, and all from within a typical ASP page. However, disadvantages of ASP are that it is very slow performance wise. Besides, it is also restricted to using only scripting languages and can't do all the things that a fully-fledged programming language can. At the same time, the scripting languages took a lot of shortcuts to make the language smaller. Some of these shortcuts make their programs longer and more complicated than is otherwise necessary.

3.4.1.2.4 ASP.NET

ASP 3.0 is the latest version of ASP, but it is hardly similar to ASP because ASP .NET is the next generation ASP. ASP .NET is an entirely new paradigm for server-side ASP scripting.

ASP .NET is a part of the new .NET (dot net) Framework. Microsoft spent three years rewriting ASP .NET from the ground up, and ASP .NET is not fully backward compatible with ASP 3.0.

3.4.1.2.5 Comparison on ASP and ASP.NET

ASP .NET has better language support, a large set of new controls and XML based components, and better user authentication. Beside that, ASP .NET provides increased performance by running compiled code. ASP .NET code is not fully backward compatible with ASP.

ASP.NET has many advantages compared to ASP, such as:

- **Better Language Support**

  ASP .NET uses the new ADO .NET and supports full Visual Basic, as well as C# (C sharp) and C++. Besides, it also supports JScript as before.

- **Programmable Controls**

  ASP .NET contains a large set of HTML controls. Almost all HTML elements on a page can be defined as ASP .NET control objects that can be controlled by scripts. Beside that, ASP .NET also contains a new set of object oriented input controls, like programmable list boxes, validation controls and a new data grid control supports sorting, data paging, and everything which are expected from a dataset control.

- **Event Driven Programming**

  ASP.NET is much more event-driven, with the event handlers running on the server. In contrast with linear program which move in a linear fashion from step 1 to step 2 and so on, event-driven programming responds to events. All ASP .NET objects on a Web page can expose events that can be processed by ASP .NET code. Load, click and change events handled by code makes coding much simpler and much better organized.

- **User Authentication, with Accounts and Roles**

  ASP .NET supports forms-based user authentication, including cookie management and automatic redirecting of unauthorized logins. Anyway, custom login page and custom user checking still can be done using ASP.NET. Beside that, ASP .NET allows for user accounts and roles, to give each user (with a given role) access to different server code and executables.

- **Higher Scalability**

  Much has been done with ASP .NET to provide greater scalability. Server to server communication has been greatly enhanced, making it possible to scale an application over several servers.

- **Increased Performance - Compiled Code**

  The code is compiled rather than interpreted, allows for the creation of larger website which is easier to scale and maintain. Performance is greatly increased at the same time.

- **Easier Configuration and Deployment**

  Configuration of ASP .NET is done with plain text files and the configuration files can be uploaded or changed while the application is running. The user no need to restart the server and no more metabase or registry puzzle. Beside that, it is easier for deployment where there are no more server restart to deploy or

replace compiled code. ASP .NET simply redirects all new requests to the new code.

- **Not Fully ASP Compatible**

  ASP .NET is not fully compatible with earlier versions of ASP, so most of the old ASP code will need some changes to run under ASP .NET. ASP .NET uses a new file extension ".aspx" to overcome the problem. As the result, ASP .NET applications are able to run side by side with standard ASP applications on the same server.

3.4.2 Consideration Of Database

Nowadays, there are many different database management systems existing in the market, with a wide range of prices and a variety of features. Selecting a particular Database Management System (DBMS) determines the particular features and limitations present in the final system. Below are some of the criteria for selecting a DBMS:

- **Ease of Use**

  The DBMS must present a front-end to the user that is clear, consistent and intuitive. If the DBMS itself does not provide such a front-end, another component of the IMS must provide it.

- **Ease of Administration**

  The DBMS must be easy to administer. An authenticated user should be able to make changes quickly and easily to any aspect of the DBMS, even those changes not possible or allowed through the users' front-end.

- **Reliability**

  The DBMS must be able to protect its data in the face of hardware or software failure. The ACID test is often used to check a DBMS against minimum reliability standards.

- **Cost**

  The cost of the DBMS must not exceed its allocation in the project budget.

- **Security**

  The DBMS must be able to authenticate and discriminate between various levels of user privilege ranging from no access to full administrative rights.

- **Compatibility**

  The DBMS must be able to function with the other hardware and software components of the Inventory Management System.

- **Familiarity with Database**

Familiarity with database is very important because if the structure and operation of the DBMS is already known, then the time which is needed for learning the DBMS can be saved, and the designer can straight forward focused on design. Besides that, prior experience with the DBMS will enable the designer to better avoid coding errors and the associated waste of debugging time.

Below, we will look at some databases and their features, limitations and strength. The databases that we are going to have a look are Microsoft Access 2002, Microsoft SQL Server 2000 and MySQL AB MySQL 4.0.

3.4.2.1 Microsoft Access 2002

Table 3.2: Description of Microsoft Access 2002

DBMS Type	Relational file-based database
Hardware Requirements	Intel Pentium III or AMD Athlon processor, 136 MB RAM, 285 MB hard drive space
Software Requirements	Internet Explorer 5.0 and one of the following: Windows NT, Windows 2000, Windows XP
Advantages	Easy to use and administer, requires minimal knowledge of SQL, imposes relatively light load on CPU, inexpensive, relatively well-suited for XML

Disadvantages	Access databases use a proprietary format mostly incompatible with SQL, cannot perform point-in-time recovery, cannot log database activity, slower than other databases, completely inaccessible to Java-based applications
Database Limitations	Database only runs on Windows, limited to 255 users, limited to 32,768 objects in database, file size limited to 2 GB
Reliability	The database file itself is much more stable than the program. Fortunately, the program does not have to be run if the file is accessed through the Web.

3.4.2.2 Microsoft SQL Server 2000

Table 3.3: Description of Microsoft SQL Server 2000

DBMS Type	Transactional relational database server
Hardware Requirements	Intel Pentium II or AMD K6-II processor, 64 MB RAM, 380 MB hard drive space
Software Requirements	Windows 2000 Server or Windows .NET Server 2003, Internet Explorer 5.0
Advantages	More reliable than one might expect from

	Microsoft, supports enterprise-class reliability and security features, compatible with many third-party application servers, can run multiple databases on one server
Disadvantages	Expensive, requires a Windows 2000 Server, can be difficult to administer
Database Limitations	Limit of approximately 2 billion objects in database
Reliability	Supports failover clusters, point-in-time recovery and other enterprise-class reliability features, can automatically restart itself if stopped.

3.4.2.3 MySQL AB MySQL 4.0

Table 3.4: Description of MySQL AB MySQL 4.0

DBMS Type	Relational database server (Transactional with InnoDB drivers)
Hardware Requirements	Minimum required for operating system
Software Requirements	Win32 or Linux/UNIX-based operating system, MyODBC (for ODBC driver support), Connector/J (for JDBC driver support)
Advantages	Free, extensive online documentation, compatible with many different operating systems

Disadvantages	Not fully SQL92 complaint (which may cause problems with application servers that form their own SQL queries), free version offers no tech support, version 4.0 and the MyODBC, Connector/J and InnoDB drivers are all in beta testing, cannot lock objects below the table level, InnoDB offers transactional support but without ACID compliance, some extensions are not compatible with the SQL99 standard
Database Limitations	Limited to 32 indexes per table, database size limited to maximum file size of operating system.
Reliability	Open-source software, appears to be almost constantly in beta, Connector/J has a history of reliability issues.

3.4.3 Consideration of Data Access Tools

3.4.3.1 ODBC

Open Database Connectivity (ODBC) is the first data access technology that provided a common standard interface through SQL for accessing heterogeneous relational databases. With ODBC, an application can access various databases through a single set of common code. The developers just need to add ODBC drivers to connect the application to the user's choice of data. Today ODBC is available across more than a

dozen platforms and a multitude of databases. The general architecture is shown in Figure 3.2:

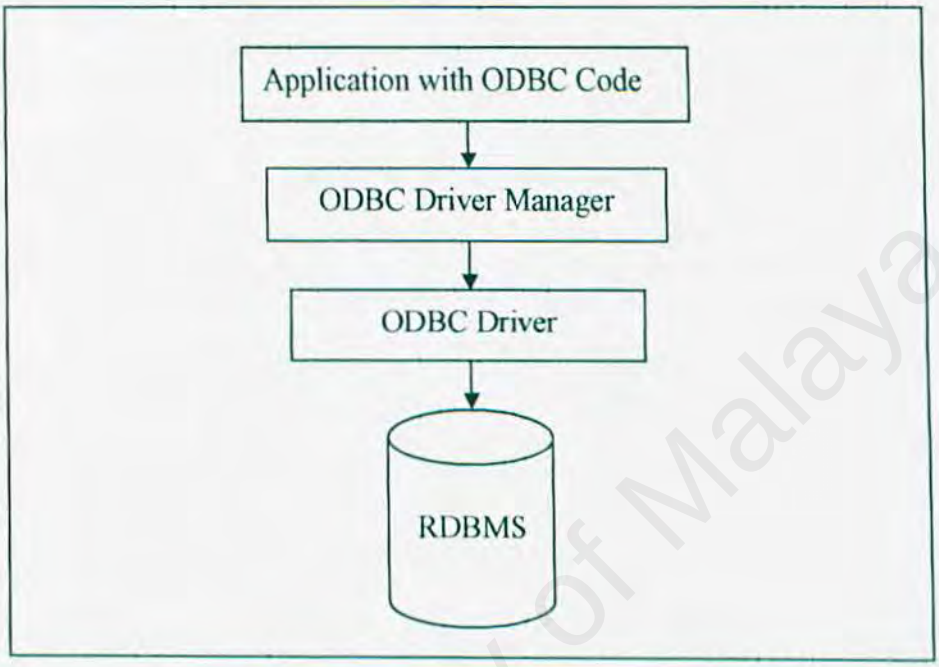

Figure 3.2: ODBC Architecture

3.4.3.2 OLEDB

Over the years, ODBC worked well in connecting to relational databases. As more and more data is stored in non-relational format, as Microsoft Exchange Server does, a new architecture was needed to provide seamless connectivity across various data sources and applications. OLE DB was the fundamental Component Object Model (COM) building block for storing and retrieving records and provides consistent access to relational databases, non-relational, and unstructured data sources across the enterprise. OLE DB

was introduced to build upon the success of ODBC. The architecture is shown in Figure 3.3:

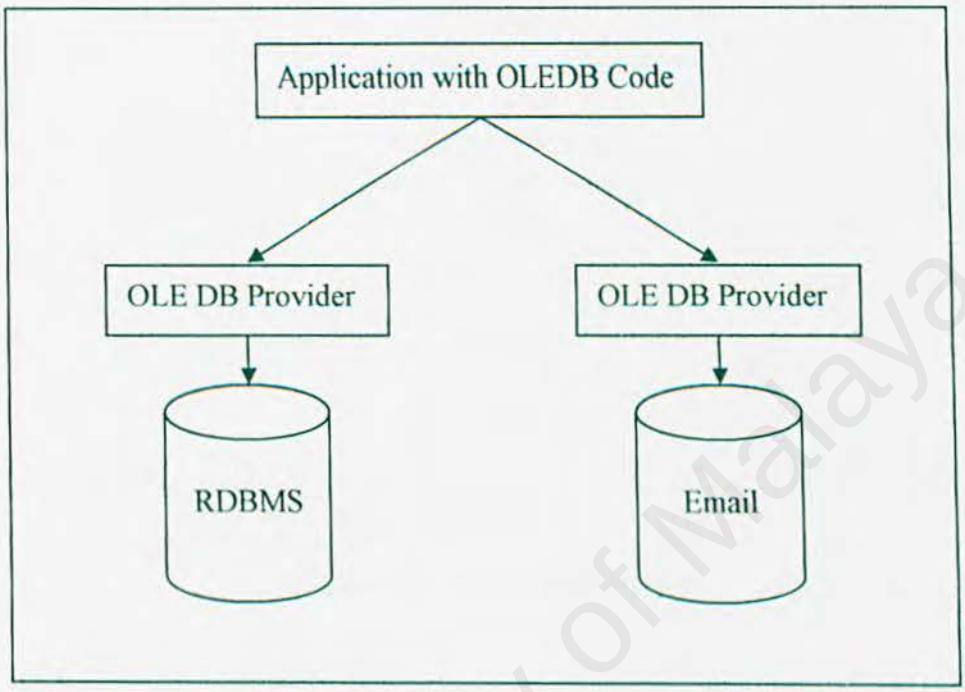

Figure 3.3: OLEDB Architecture

3.4.3.3 ADO

OLE DB utilizes several technical low-level calls that can make programming tedious, especially for Visual Basic type of programs. To overcome this problem, Microsoft introduced ActiveX Data Objects (ADO). ADO works on top of OLE DB providing a simpler, higher-level interface, as shown in Figure 3.4:

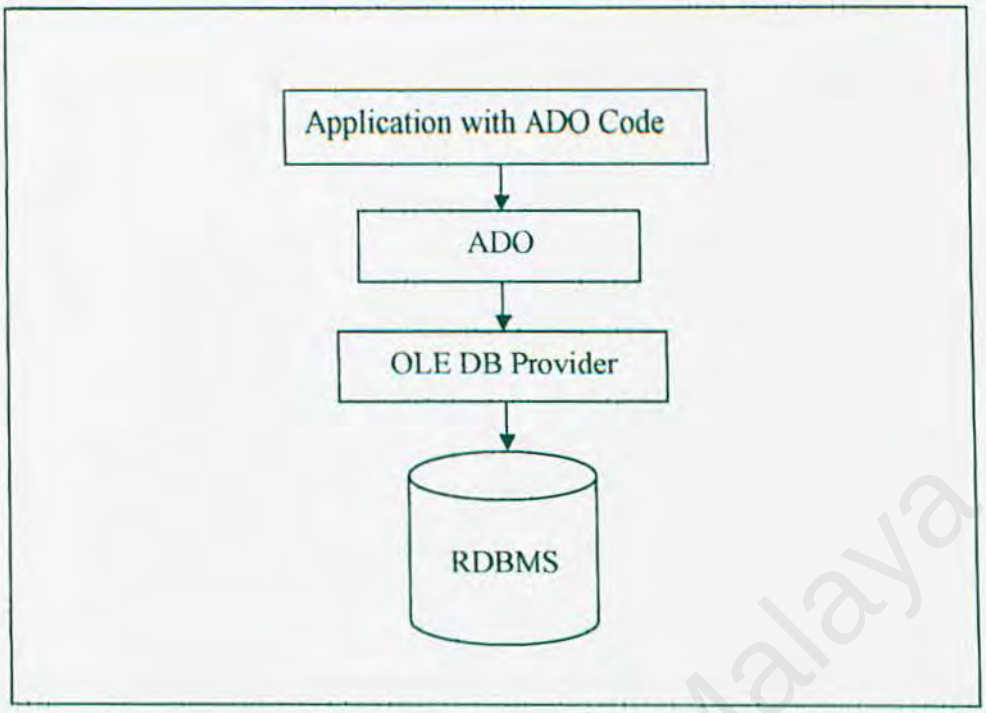

Figure 3.4: ADO Architecture

3.4.3.3 ADO.NET

Microsoft decided to redesign the data access model when it started designing the .NET architecture. Rather than extending ADO further, Microsoft decided to design a new data access architecture based on the new .NET framework, but, it still kept the acronym ADO involved anyway. Microsoft designed ADO.NET based on its experience with its successful ADO object model, but with a completely different architecture based on XML and a disconnected computing model. The new architecture provides several ways an application can connect to a data source as shown in Figure 3.5:

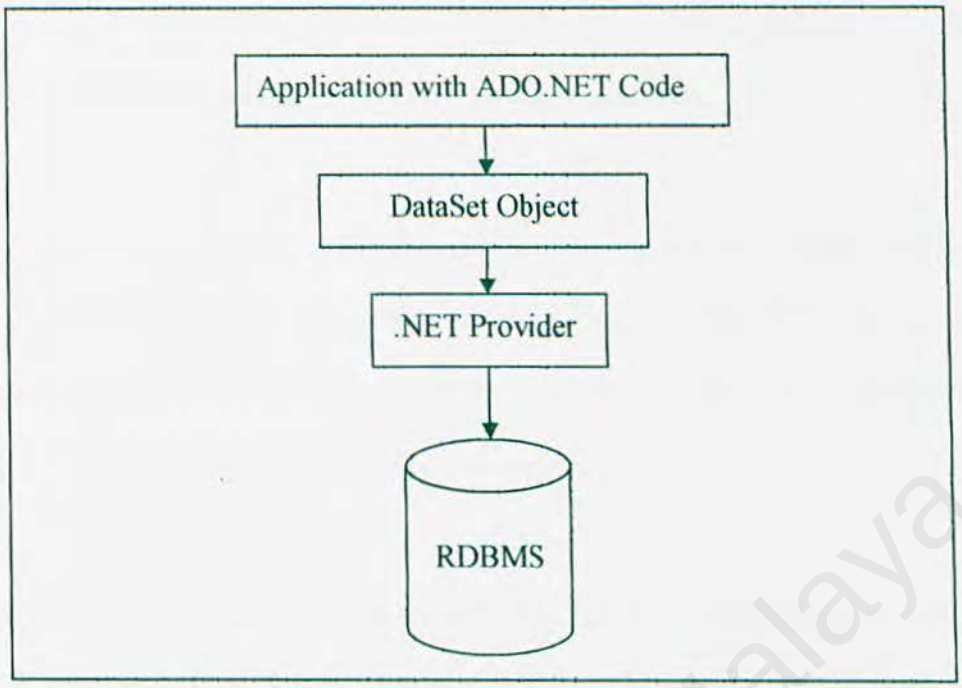

Figure 3.5: ADO.NET Architecture

3.4.3.5 Comparison between ADO and ADO.NET

The main difference between ADO and ADO.NET is ADO.NET addresses three important needs that ADO does not address, such as

- Providing a comprehensive disconnected data-access model, which is crucial to the Web environment.
- Providing tight integration with XML
- Providing seamless integration with the .NET Framework (for example, compatibility with the base class library's type system).

From an ADO.NET implementation perspective, the Recordset object in ADO is eliminated in the .NET architecture. ADO.NET has several dedicated objects led by the DataSet object and including the DataAdapter and DataReader objects to perform

specific tasks. Besides that, ADO.NET DataSets operate in disconnected state whereas the ADO RecordSet objects operated in a fully connected state.

A completely new technology is needed in ADO.NET for accessing data from the .NET platform because of the fundamental differences in COM and .NET (and therefore fundamental difference in ADO and ADO.NET architecture). In fact, this eliminates the need for the existing ADO data provider technology.

A COM-based ADO data provider would have been developed in C++ with COM libraries, where as the ADO.NET data provider is required to be written with only the .NET Framework base class libraries and is designed to work completely within the parameters of Common Language Runtime (CLR). Besides, the ADO.NET environment is designed by default to work with data in XML format rather than native OS types of ADO.

3.4.4 Consideration of Web Server

A Web server is a program that, using the client/server model and the World Wide Web's Hypertext Transfer Protocol (HTTP), serves the files that form Web pages to Web users (whose computers contain HTTP clients that forward their requests). Every computer on the Internet that contains a Web site must have a Web server program.

There are many criteria that should be considered when choosing a web server, some of them are listed below:-

- **Ease of Use**

  The web server must present a front-end to the user that is clear, consistent and intuitive.

- **Ease of Administration**

  The web server should be easy to administer.

- **Ease of Programming**

  Different web servers utilize different programming languages in which some of them are easier to learn, use and maintain. An easier language results in a faster design time as well as easy in learning.

- **Cost**

  The cost of the web server must not exceed its allocation in the project budget. Some of them are free such as Apache.

- **Compatibility**

  The web server must be compatible with the database management system, DBMS.

- **Familiarity with Web Server and Programming Language**

  If the developers or designers are familiar with the structure and operation of the web server and its associated programming language, then they can save their time of learning the server technology and focus toward design. Besides that, prior experience with the server technology will enable the designer to better avoid coding errors and the associated waste of debugging time.

Below are some of the web servers that are available nowadays.

3.4.4.1 Jakarta Tomcat 4.1

Table 3.5: Description of Jakarta Tomcat 4.1

Programming language	JSP
Hardware Requirements	Minimal
Software Requirements	Win32 or Linux-based operating system, J2SE Development Kit (JDK 1.3) for operating system chosen.
Advantages	Free, low server overhead
Disadvantages	Requires specialized Java compiler, no graphical administration tools, requires in-

	depth knowledge of Java to program, does not appear to support web servers other than Apache.
Compatible databases	Any database with Type IV JDBC drivers
Reliable	Unknown

3.4.4.2 Macromedia ColdFusion MX

Table 3.6: Description of Macromedia ColdFusion MX

Programming language	JSP, CFML
Hardware Requirements	Intel Pentium processor, 512 MB RAM, 400 MB hard drive space
Software Requirements	One of the following: Windows NT/2000/XP, Red Hat Linux 6.2-7.2, SuSE Linux 7.2-7.3, Turbolinux 8 Server
Advantages	Platform-independent, extremely compatible, requires no specialized coding environment, development using Dreamweaver MX requires no knowledge of Java or SQL and only basic knowledge of CFML, easy-to-use Web-based administration, J2EE-compliant, CFML language is very easy to learn, ColdFusion MX Server Developer Edition is a

	free download, can interface with XML, ASP.NET, SOAP and other web services, built-in XML and XSL handling, can import JSP functions and convert them to CFML.
Disadvantages	Expensive, does not provide a production-environment web server, not fully compatible with MySQL.
Compatible databases	Any database with Type III/IV JDBC drivers
Reliable	ColdFusion MX Server Developer Edition has run on Sean's home computer for 62 days continuously at the time this was written, without noticeably affecting the performance of other applications.

3.4.4.3 Microsoft Internet Information Services (IIS) 5.1

Table 3.7: Description of Microsoft Internet Information Services (IIS) 5.1

Programming language	ASP, ASP.NET
Hardware Requirements	None above those of host operating system
Software Requirements	Windows 2000 Server
Advantages	Integrated web server, relatively easy to administer, extensive XML support
Disadvantages	Platform-specific, only runs securely on

	Windows 2000 Server, ASP.NET applications require clients to install the .NET Framework, ASP and ASP.NET typically require more lines of code than Java or CFML to accomplish a given task.
Compatible databases	Microsoft SQL Server 2000, any database with ODBC drivers
Reliable	Not terrible, assuming the underlying Windows 2000 Server is properly configured.

3.4.5 Consideration of Multimedia Tools

3.4.5.1 Macromedia Flash

Flash is a vector based animation program and it is currently a representative product of Macromedia for web-based multimedia. With a light plugin (for the latest version the plugin is under 300Kb - that's less than 2 minutes download time over a 28.8 Kbps modem) and light movies, it often running less than 20 Kb and present a broadcast quality presentation experience. The graphics created within Flash are not bitmap based as JPEG and GIF graphic; but the images created inside of Flash are created mathematically where point "A" is plotted by Flash to point "B" and "C" and the color in between is filled in. Anyway, Flash has a built in graphics editor that done this.

The core functionality for Flash has always been animation but not the blocky animation associated with Animated GIF's. Control is the focus for Flash's animation.

When a movie is created within Flash the user can manage events in the movie with timelines, scenes and ActionScript. These tools manage how the movie is played back on the web page. ActionScript is a scripting language with is used by Flash to glue elements within a movie together the same way JavaScript glues HTML elements together in a web page.

3.4.5.2 Macromedia Director

Macromedia Director builds rich, interactive presentations as Flash but they are not doing double duty. Director bridges the gap between the Web and CD-ROM development because the strong point for Flash is web creation. The main types of graphics used in Director are bitmap and it allows for the support of Xtra's. An Xtra is a program in of itself. It has built in functionality, such as the ability to play back Flash movies within a Director movie without requiring the user to have the Flash plugin. There are hundreds of Xtra's for Director that allows the user to do almost anything but it must be used witj care on the web. The scripting language for director is Lingo and it can do much more than ActionScript. For the new 3D control in Director 8.5 there are over 400 new Lingo functions, while ActionScript barely has 300 for the entire program environment.

3.4.6 Selection for Development Tools

- **Web Programming Language – ASP.NET**

It has been chosen because of its easy development, compatibility, high scalability, as well as better language support. Besides that, it is much more event-driven, with compiled code instead of being interpreted.

- **Database – Microsoft Access 2002**

  It has been chosen because it is easy to use and administer, inexpensive, it is stable and the developer is familiar with it.

- **Data Access – ADO.NET**

  It has been chosen because it provides a comprehensive disconnected data access model and seamless integration with the .NET framework.

- **Web Server – Microsoft Internet Information Services (IIS) 5.1**

  It has been chosen because it can support the chosen programming language – ASP.NET and it is relatively easy to administer.

- **Multimedia Tools – Macromedia Flash**

  It has been chosen because it is more suitable to use for developing an interactive website.

3.5 Hardware Requirements

3.5.1 Development Requirements

- System Type- x86- based PC with Windows XP
- Processor- Pentium III 733Mhz
- Memory- 256MB
- Hard disk space – 20GB

3.5.2 Deployment Requirements

3.5.2.1 Server Hardware Requirements

- IBM compatible PC with Windows XP
- Processor- Pentium IV 2Ghz
- Memory- 512MB
- Hard disk space – 30GB

3.5.2.2 Client Hardware Requirements

- IBM compatible PC
- Pentium II 300 MHz and Above
- Memory: 64 MB

Chapter 4 – System Analysis and Design

System design is defined as those tasks that focus on the specification of a detailed computer-based solution (Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, 2001). It is an important part of software development process and focuses on the technical or implementation concerns of the system. Typical phases in design process are architectural design, abstract specification, interface design, component design, data structure design and algorithm design.

4.1 Client / Server Architecture Model

The client-server architecture model is a distributed system model which shows how data and processing are distributed across a range of processors (Ian Sommerville, 2001). It describes the relationship between two computer programs in which one program (the client) makes a service request from another program (the server) which fulfills the request. Client-server architecture model is an important idea in a network where it provides a convenient way to interconnect programs that are distributed efficiently across different locations.

4.1.1 Three-Tier Client Server Architecture

Inside the three-tier client-server architecture, the three components of an application - presentation, processing and data are divided into three distinct software entities (tiers). It is emerged to overcome the limitations of the two-tier clients-server architecture in

which a middle tier was added between the user system interface client environment and the database management server environment. There are a lot of ways of implementing the middle tier, such as application servers, transaction processing monitor and message servers. The three-tier client-server architecture for this website is show at Figure below. It consists of client-tier (user interface that runs on the user's computer), application-server-tier (modules that process data and protect the data from direct access by the clients) and data-server-tier (database management system, DBMS that stores the data required by the middle-tier).

Advantages of three-tier client-server architecture model are as below:-

- Flexible resource allocation in which the middle-tier servers are portable and can be dynamically allocated and shifted according to needs
- Separate software entities allowed for parallel development of individual tiers by the application specialist which will increase the overall quality
- Modularly designed middle tier code modules can be re-used by several applications

4.1.2 The Three-Tier Client-Server Architecture in this website

The three-tier client-server architecture for this website is implemented as shown in Figure below.

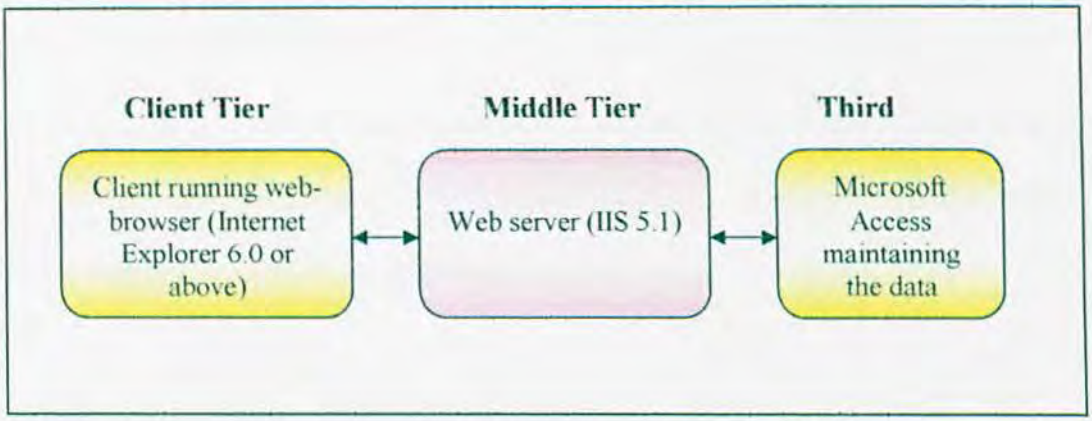

Figure 4.1: Three-tier client-server architecture of the website

- **Client-tier**

  Client-tier is constituted of computers with Internet Explorer (4.0 or above). Clients process their application and manipulate their data through the user interfaces.

- **Middle-tier**

  Middle-tier consists the web server, IIS 5.0 in this system. The web server processes the request from the client and returns the required result in web pages format. The web server is linked to database for processing the required data.

- **Third-tier**

  It contains the database server, Microsoft Access to maintain the data records.

4.2 Database Design

A database is a collection of interrelated files. Database of this website is used to store all the information of the history of AI. Database management system, DBMS is used to access, control and manage the database.

4.2.1 Entity Relationship Diagram (ERD)

Data modeling or database modeling is a technique for organizing and documenting a system's data. Data model of this website is represented using an entity-relationship diagram or ERD.

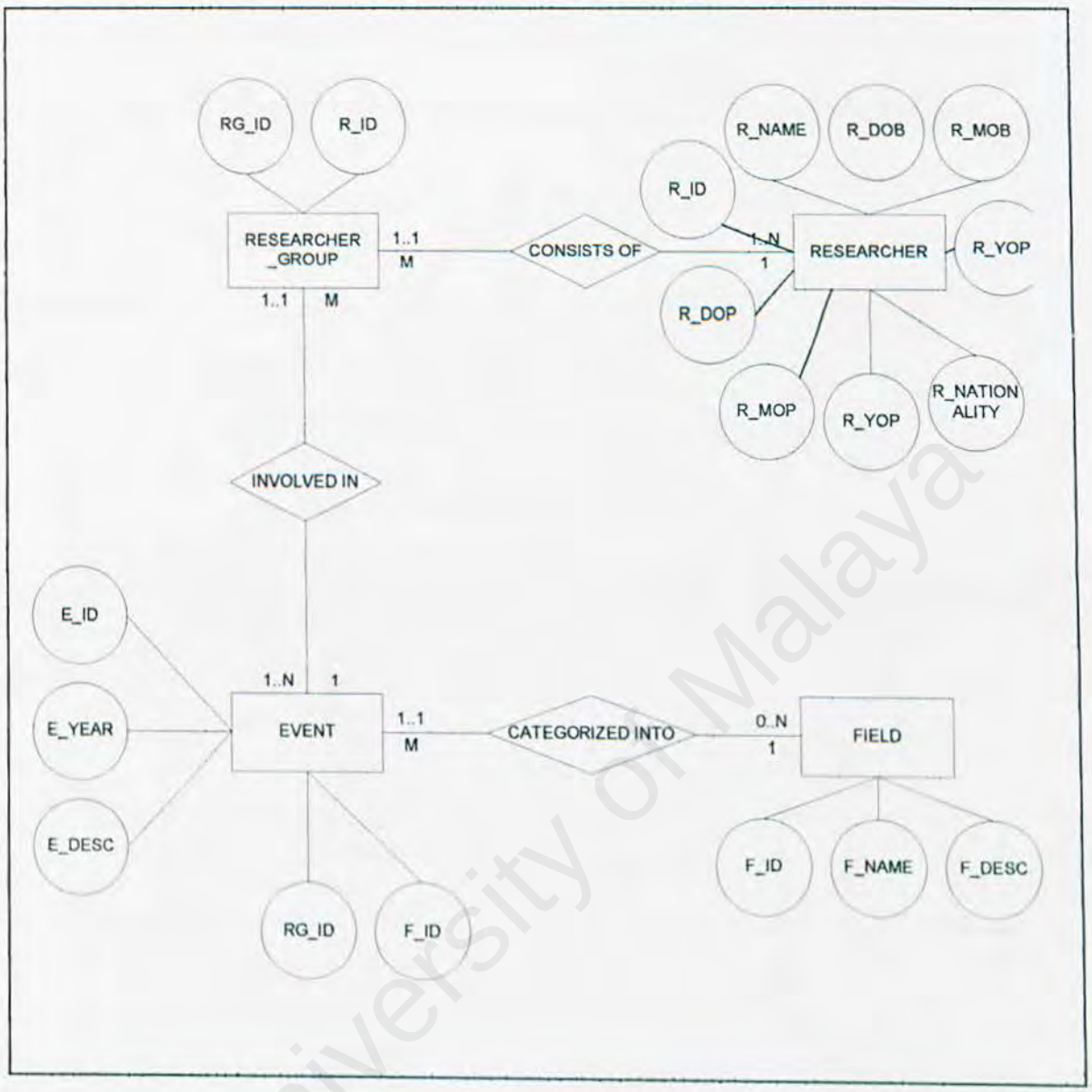

Figure 4.2: ER Diagram for the website

4.2.2 Data Dictionary

Data dictionary or metadata defines the field, field type and descriptions of each table inside the database.

Researcher

This table stores each of AI researcher's information.

Table 4.1: Table of Researcher

le Name	Attribute Name	Contents	Type	Format	Required	PK/FK	PK Referenced Table
ARCHER	R_ID	Unique ID for a researcher	AUTO NUMBER	999999	Y	PK	
	R_NAME	The name of the researcher	VCHAR(30)	Xxxxxxxxx	Y		
	R_DOB	The day of birth of the researcher	NUMBER	999999	N		
	R_MOB	The month of birth of the researcher	NUMBER	999999	N		
	R_MOB	The year of birth of the researcher	NUMBER	999999	N		
	R_DOP	The day the researcher passed away (blank if still alive	NUMBER	999999	N		
	R_DOP	The month the researcher	NUMBER	999999	N		

		passed away (blank if still alive					
	R_DOP	The year the researcher passed away (blank if still alive	NUMBER	999999	N		
	R_NATIONALITY	The nationality of the researcher	VCHAR(20)	Xxxxxxxxx	N		

Researcher Group

This table stores a group of researchers (from Researcher table) that involved together in an event.

Table 4.2: Table of Researcher Group

e Name	Attribute Name	Contents	Type	Format	Required	PK/FK	PK Referenced Table
ARCHER ROUP	RG_ID	Unique ID for a group of researchers	NUMBER	999999	Y	PK	
	R_ID	Corresponding researcher's ID that is in the group	NUMBER	999999	Y	FK	RESEARCHER

Field

This table stores the information about each of the fields in AI, events are categorized into fields.

Table 4.3: Table of Field

ble me	Attribute Name	Contents	Type	Format	Required	PK/FK	PK Referenced Table
ELD	F_ID	Unique ID for the field	AUTO NUMBER	999999	Y	PK	
	F_NAME	The name of the field	VCHAR(30)	Xxxxxxxxxx	Y		
	F_DESC	The description of the field	VCHAR(250)	Xxxxxxxxxx	N		

Event

This table stores the information about each of the event related to AI, with reference to its group of researchers and its field.

Table 4.4: Table of Event

e e	Attribute Name	Contents	Type	Format	Required	PK/FK	PK Referenced Table
NT	E_ID	Unique ID for the event	AUTO NUMBER	999999	Y	PK	
	E_YEAR	The year when the event took place	NUMBER	9999	Y		
	E_DESC	The description of the event	VCHAR(2000)	Xxxxxxxxxx	N		
	RG_ID	The corresponding researcher group's ID that involved in the event	AUTO NUMBER	999999	Y	FK	RESEARCHER_GROUP
	F_ID	The corresponding field's ID of the event	AUTO NUMBER	999999	Y	FK	FIELD

4.3 Program Design

The program that has been designed must satisfy user requirements where it must be easy to read and understand. It enables other person to amend it later. A variety of design tools such as structured chart and data flow diagram are helped in developing well-structured program.

4.3.1 Structured Chart

Structured chart shows the top-down design of a program where it make it easy to view the system as a whole.

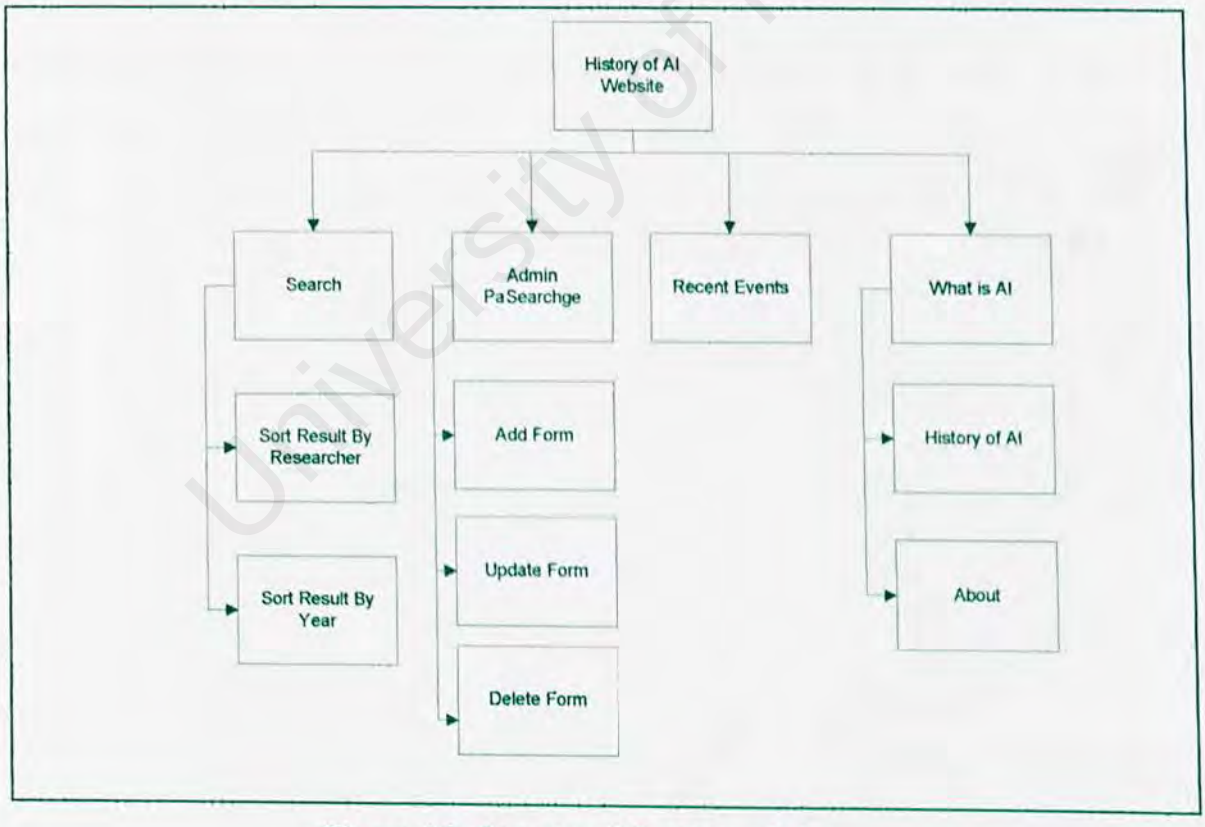

Figure 4.3: Structured for the website

In the website, there are four main modules – Introduction, Search, Browse and Recent Events. Introduction module gives an overview of Artificial Intelligence as well as introducing the website and its objectives. In Search module, users can search for information and get the result sorted according to researchers, year of the events or the field of the events. In browse module, similarly, users can browse for information according to researchers, years and fields. Recent Event module basically has all the latest or recent events in the arena of Artificial Intelligence.

4.3.2 DFD Diagram

Data flow diagram is a tool that depicts the flow of data through a system and the work or processing performed by that system (Jeffrey L. Whitten, Lonnie D. Bentley, Kevin C. Dittman, 20001). It represents the input of data to a process or the output of data from a process.

4.3.2.1 Context Diagram

The Context Diagram shows all the main process for this website as the diagram shown in figure 4.8

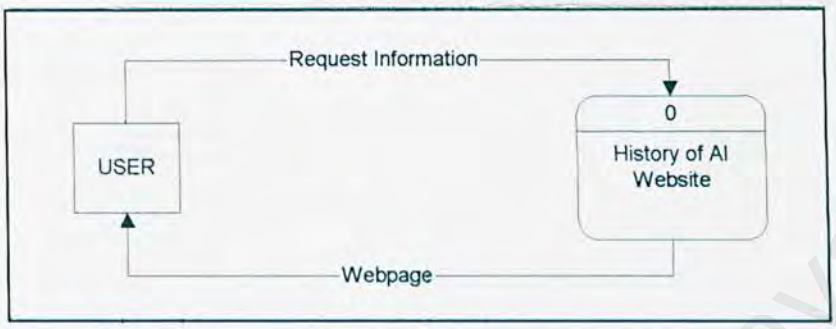

Figure 4.4: Context diagram for the website

4.3.2.2 Level 1 DFD

Level 1 Data Flow Diagram for the whole system is shown in figure 4.9 below.

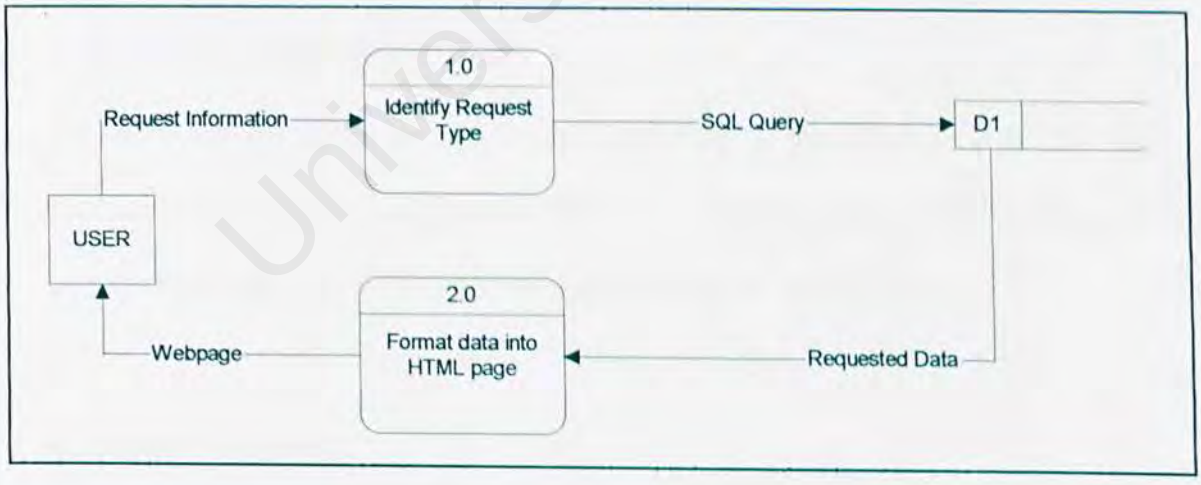

Figure 4.5: Level 1 DFD for the website

4.3.2.3 Level 2 DFD

Data Flow Diagram for the whole system is shown at figure 4.10 below.

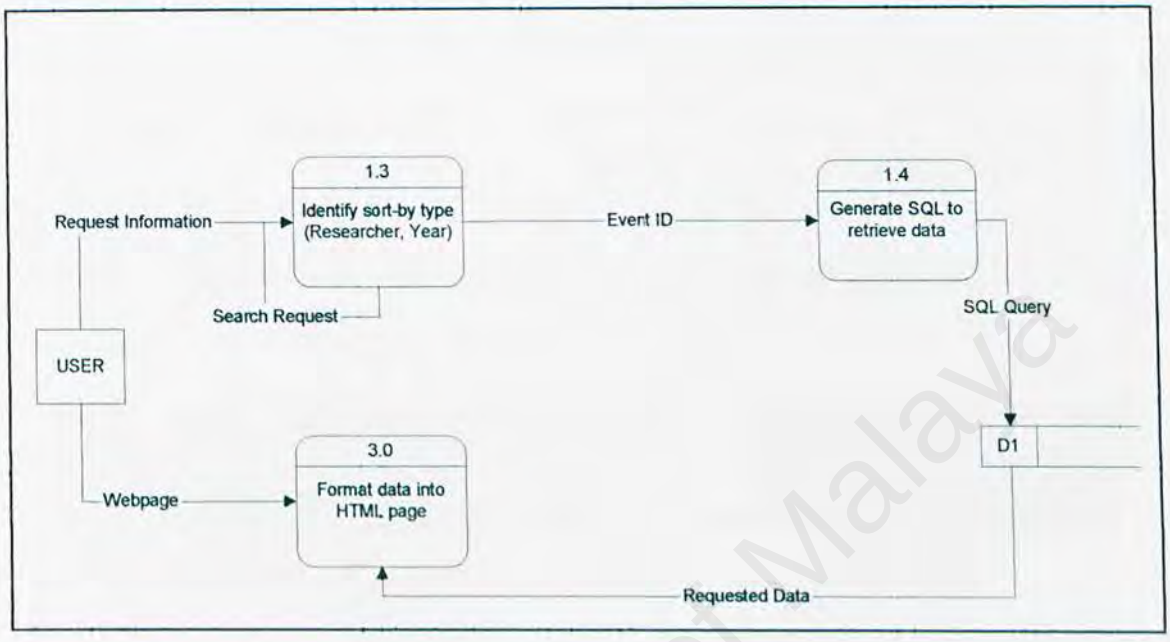

Figure 4.6: Level 2 DFD for the website

4.4 User Interface Design

The user interfaces of this website are designed based on the web page style. The user interface should be user friendly and attractive as well. It should provide the users with easy navigation through web pages and make requests on specific task.

4.4.1 Screen Design

Screen design which is user friendly is required in this website. Each interface have unique and standardize features and will be equipped with relevant links and

information. A formal design in user interface is to suites their purpose for delivering information directly and clearly. Each of the interface have its own function.

4.4.1.1 Main Page

This is the interface design for the main page.

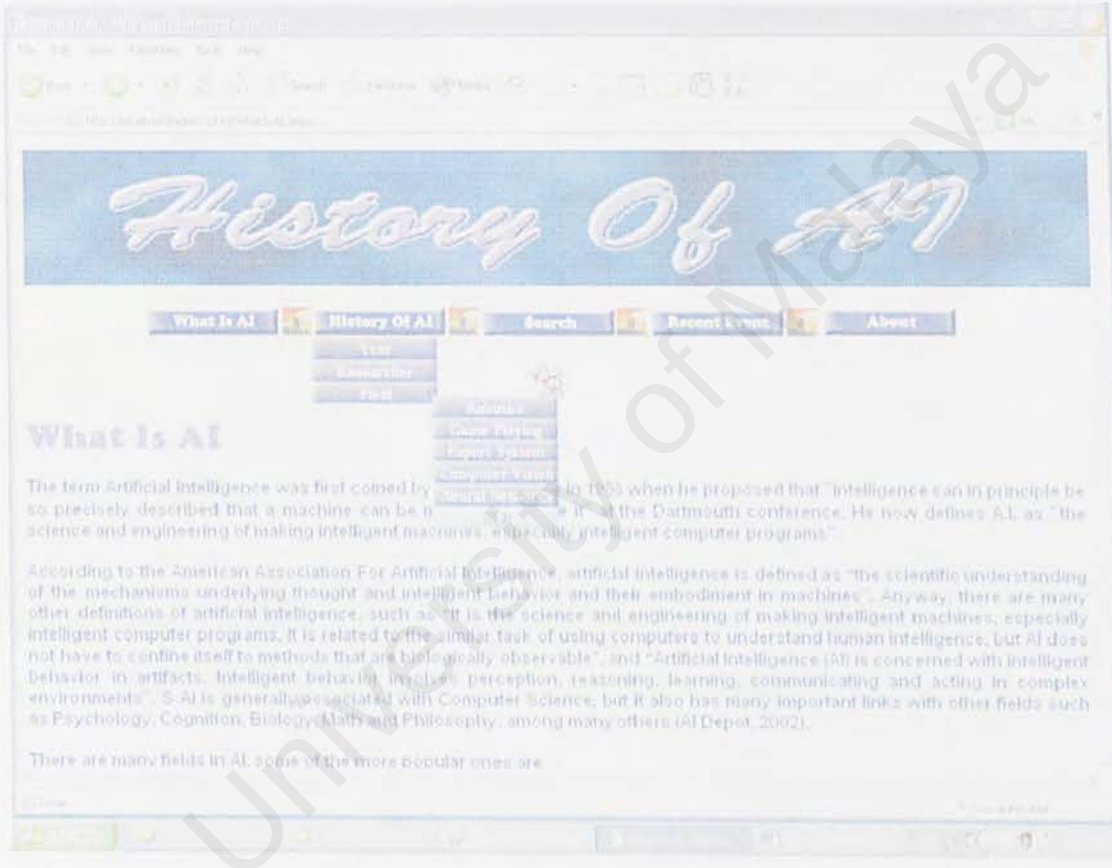

Figure 4.7: The interface design for the User's Main page

4.4.1.2 Administrator's Main Page

This is the interface design for the main page.

Figure 4.7: The interface design for the administrator's main page

4.4.1.3 Search Page

This is interface design for the search page

Figure 4.8: The interface design for the search page

4.4.1.3 Search Page

This is interface design for the game playing field

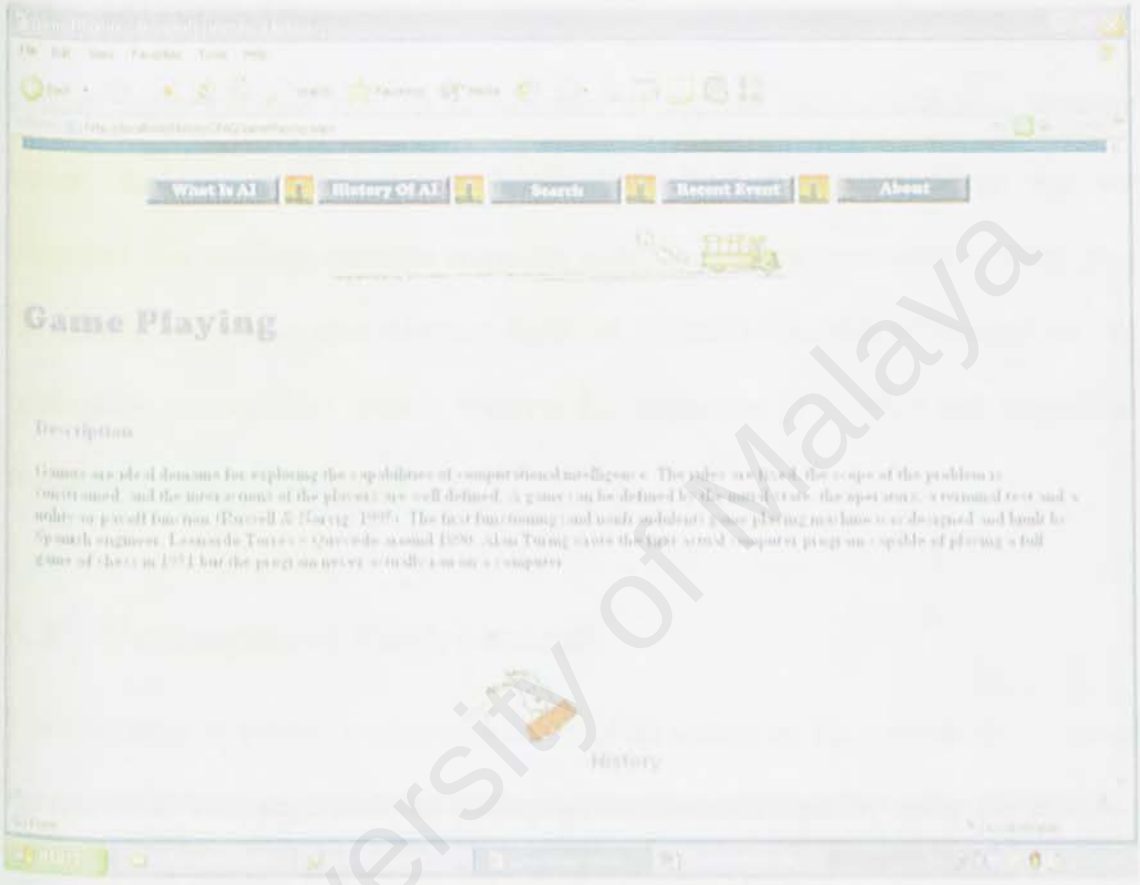

Figure 4.8: The interface design for the game playing field

Chapter 5 – System Implementation

5.1 Introduction

System implementation is a set of activities to produce this website as a working system. Requirements and system specifications from the system design step are translated into machine readable computer code. System implementation, which uses the system design as its core reference make the physical realization of the database and application designs. This process involves the integration of physical and conceptual resources, outputs and constraints.

5.2 Development Environment

The first stage of system implementation involves setting up the suitable development environment. This stage includes setting up development tools by using the dynamic hardware and software to facilitate the system implementation.

5.2.1 Hardware Development Environment

All the hardware that are needed must be prepared and tested out. Below is the list of the hardware configuration of this website development environment: -

- Desktop – Dell Computer Corporation OptiPlex GX110
- BIOS- Dell Computer Corporation
- Dell Compatible Computer – 256 MB RAM, Intel
- Pentium III 733MHz processor

- CD-ROM driver
- Monitor
- 20 GB Hard Disk
- Standard floppy disk drive, printer, and standard modem for internet connection

5.2.2 Software Development Environment

The installation and the configuration of the software and development tools need to be done before the actual coding activities starts.

5.2.2.1 Software Tools for Design and Documentation

Microsoft Word 2002, Microsoft Visio 2000, and Microsoft Project Professional 2002 were used as software tools in system design and documentation to draw the Entity Relationship diagram (ER diagram), structure chart and data flow diagrams (DFD diagrams).

5.2.2.2 Software Tools for the Prototype Development

The software tools, which used in developing the website, are shown in the Table 5.1.

Table 5.1: The Software Tools for The Development Of The Website

Software	Usage	Description
Microsoft Windows XP Server	Development Environment System Requirement	Operating System
Microsoft Internet Information Services (IIS) 5.1	Web Server	
Microsoft Access 2002	Database Design	Database System
Macromedia Dreamweaver MX	System and Interfaces	Solution for web-based

	Design	design
Macromedia Fireworks MX	Interface Design	Solution for web-based design
Adobe Photoshop7.0	Interface Design	Solution for web-based design
Microsoft Internet Explorer 6.0	Web Browser	

5.3 Database Development

The website used Microsoft Access 2002 as the database server. Microsoft Access 2002 Database Server was installed. An empty database called AI was created. Entities, attributes and relationships are defined and tables are created by specifying all the fields and the properties of the fields. Any data creation, updates or data retrieval will be connected directly to the database server through ODBC.

The development of the database is a process to transform the database design into tables based on the logical data model of the website during the system design phase. The process of database table design is an iterative process where it changed from time to time to fulfill the requirements.

The newly discovered tables will be added. At the same times, all the unnecessary tables were eliminated from database to avoid data overlapping and to reduce workload of the entire system when deployment.

5.4 Web Application Development

Code generation that translates all the algorithms into ASP.NET programming language instruction is important task in application development.

Several programming principles need to be used in the program to ensure the consistency, maintainability and reliability of the system. The principle is included:-

a) Meaningful Variables names, procedure name and parameter variable.

b) Declarations must is a stand out condition to increase the readability.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>
<html>
<head>
<title>Remove Event</title>

<script language="C#" runat="server">
    // Variable Declaration                                    → Declaration
    OleDbCommand myCmd = new OleDbCommand ( );
    OleDbConnection myConn = new OleDbConnection (
                    "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
                    "F:\\Inetpub\\wwwroot\\HistoryOfAI\\database\\AI.MDB" +
                    "; Persist Security Info=False" );
    String id;

    protected void Page_Load ( object src, EventArgs e ) → Comment
    {
      // Initialization
      myCmd.Connection = myConn;                    → Meaningful
      id = Request.QueryString [ "id" ] ;             Variable
    }                                                 Name
</script>
```

Figure 5.1: Sample Code of Deleting an Event

5.4.1 Web Server

(a) IIS (Internet Information Services) Web Server v5.1, with .NET framework

Internet Information Services (IIS) is a comprehensive web server by Microsoft. IIS version 5.1 comes together with Microsoft XP Professional operating system. IIS v5.1 therefore can be installed simply by inserting Microsoft XP Professional installation CD to the CD-ROM, go to Control Panel → Add/Remove Programs → Add/Remove Windows Components and tick on the "Internet Information Services (IIS)" checkbox However with IIS v5.1 is not sufficient to hold ASP.NET web pages, Windows .NET framework need to be installed using Windows Component Update CD. Once all these are installed, every other things like setting website securities will be as easy as a few mouse clicking since all these are integrated into the operating system.

(b) The Root Directory

A physical directory which labeled as website root directory is created in operating system drive/partition after installing IIS v5.1. All the ASP.NET pages need to be placed in *c:\Inetpub\wwwroot\HistoryOfAI*, where in this case the operating system is installed in drive *c:*.

654.2 Programming Language – ASP.NET with C#

(a) The Tool – Macromedia Dreamweaver MX

Macromedia Dreamweaver MX is a professional HTML editor for designing, coding, and developing websites, web pages, and web applications. Either by using

hand-coding or its visual editing environment, Dreamweaver MX provides helpful tools to make web creation fast and efficient.

The visual editing features in Dreamweaver enable coders to quickly create web pages with least hand coding. In spite of that, Dreamweaver also includes many coding-related tools and reference for ease of coding. Beside supporting ASP.NET and C# for building dynamic database-backed web applications, other server languages supported are ASP, ColdFusion Markup Language (CFML), JSP, and PHP.

(b) The Working Space

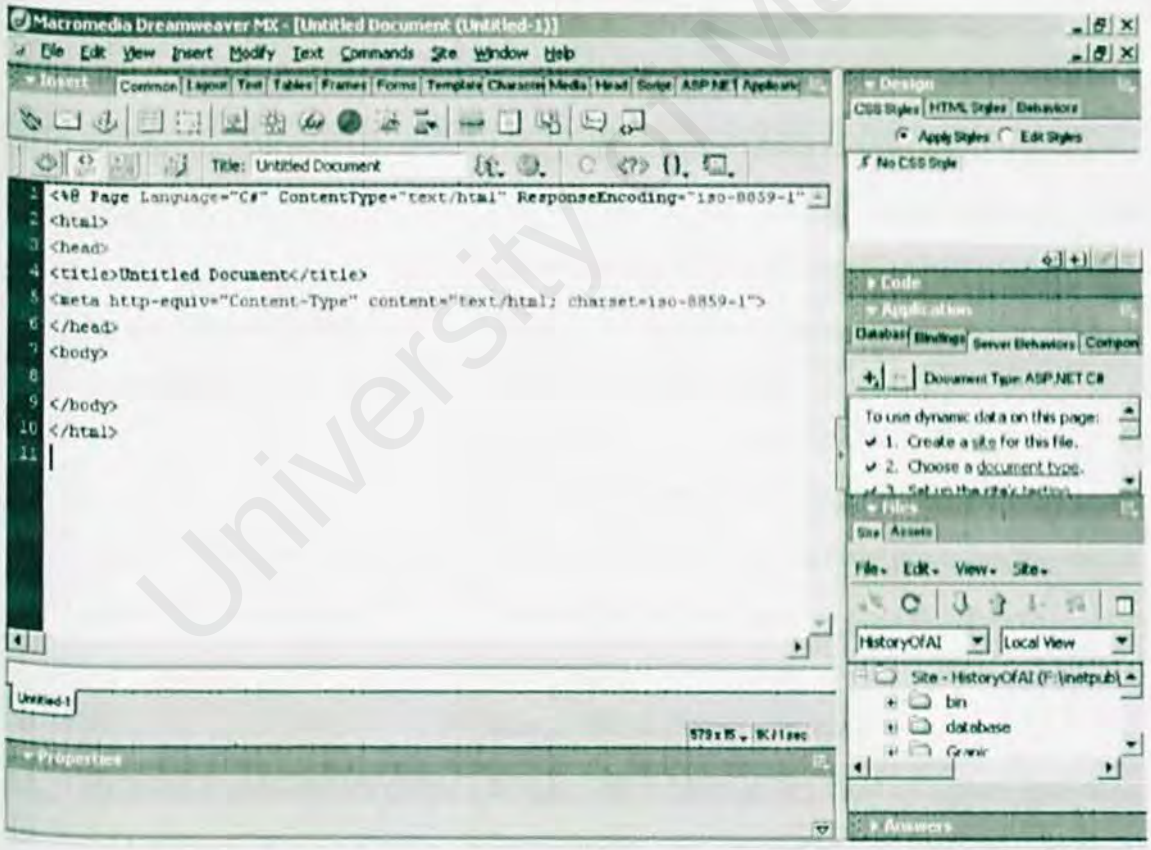

Figure 5.2: Show The Working Space For Dreamweaver MX.

(c) The Method

Below are the main files of the system:

Table 5.2: Main File Inside The system

No.	File Name	Description
1.	WhatIsAI.aspx	Brief description about AI. Have hyperlink that link to other files.
2.	Event.aspx	Display the history of AI by year.
3.	Researcher.aspx	Display the history of researcher in AI.
4.	Robotics.aspx	Description of the field of robotics. Have hyperlink that link to HistoryRobotics.aspx.
5.	HistoryRobotics.aspx	Display the history of the field of robotics.
6.	GamePlaying.aspx	Description of the field of game playing. Have hyperlink that link to HistoryGamePlaying.aspx.
7.	HistoryGamePlaying.aspx	Display the history of the field of game playing.
8.	ExpertSystem.aspx	Description of the field of expert system. Have hyperlink that link to HistoryExpertSystem.aspx.
9.	HistoryExpertSystem.aspx	Display the history of the field of expert system.
10.	ComputerVision.aspx	Description of the field of computer vision. Have hyperlink that link to HistoryComputerVision.aspx.
11.	HistoryComputerVision.aspx	Display the history of the field of computer vision.
12.	NeuralNetworks.aspx	Description of the field of neural networks. Have hyperlink that link to HistoryNeuralNetworks.aspx.
13.	HistoryNeuralNetworks.aspx	Display the history of the field of neural networks.
14.	NLP.aspx	Description of the field of natural language processing. Have hyperlink that link to HistoryRobotics.aspx.
15.	HistoryNLP.aspx	Display the history of the field of natural language

		processing.
16.	Search.aspx	Search the website.
17.	Result.aspx	Display the result of the search page.
18.	RecentEvent.aspx	Display the most recent history of AI (>= year 2000)
19.	About.aspx	Brief description about the website. Administrator can login through this page.
20.	Login.aspx	Let administrator to login into the administrator's main page.
21.	Login.ascx	A login web control.
22.	AddEvent.aspx	Add form for adding new event of history of AI.
23.	AddField.aspx	Add form for adding new field of history of AI.
24.	AddResearcher.aspx	Add form for adding new researcher of history of AI.
25.	UpdateEvent.aspx	Update form for updating existing event in the database.
26.	UpdateField.aspx	Update form for updating existing field in the database.
27.	UpdateResearcher.aspx	Update form for updating existing researcher record in the database.
28.	DeleteEvent.aspx	Delete event from the database.
29.	DeleteField.aspx	Delete field from the database.
30.	DeleteResearcher.aspx	Delete researcher record from the database.
31.	AdminMain.aspx	Administrator's main page.
33.	Header.aspx	Header that includes in most of the file.

Expert.aspx is basically display the brief description of the field of expert system. The paeg have one hyperlink that link to another file (history of expert system). The file includes the header which was a dropdown menu that built by using Macromedia Fireworks MX.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>

<html>
<head>
<title>Expert System</title>
<!-- #include file="C:\Inetpub\wwwroot\HistoryOfAI\Header.aspx"-->
</head>

.... ....

<script language="C#" runat="server">
  protected void Page_Load ( object src, EventArgs e ) {
        OleDbConnection connGlobal = new OleDbConnection (
      "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
      Server.MapPath ( "~/database/AI.mdb" ) );

    // define the command query
    OleDbCommand myCmd = new OleDbCommand ( "select F_DESC from
FIELD where F_ID = 3", connGlobal );
    // open the connection
    connGlobal.Open ( );

    // bind the datalist to the reader
    myList.DataSource = myCmd.ExecuteReader (
      CommandBehavior.CloseConnection );
    myList.DataBind ( );

    // close the connection
    connGlobal.Close ( );
  }
</script>

<body background="Image/background1.gif">
<p align="center"><img src="Grapic/bar001.gif" width="407" height="47"></p>
<p><strong><font color="#3300ff" size="6" face="cooper black">Expert
System</font></strong></p>
```

Include file

```
<hr size=3 width=90%>

<p align="left">

<asp:datalist id="myList" repeatcolumns=1 runat="server">

  <itemtemplate>
    <table cellpadding=10>
    <tr>
    <td valign="top" align="justify">
        <b><strong><font color="#0000CC"
size="4">Description</font></strong></b><br><br>
                    <strong><font color="#000066" size="3"><%# ( (
IDataRecord ) Container.DataItem ) [ "F_DESC" ] %></font></strong><br>
    </td>
          </tr>
    </table>
  </itemtemplate>

</asp:datalist>

</p>

<hr size=2 width=90%>

<p align="center"><img src="Grapic/cart0021.gif" width="131"
height="123"><strong><font size="4" face="Arial, Helvetica, sans-serif"><a
href="HistoryExpertSystem.aspx">
 History</a></font></strong></p>

</body>
</html>
```

Figure 5.3: Sample Code Of ExpertSystem.aspx

HistoryExpertSystem.aspx displays the history of the field of expert system. The data are put inside data repeater.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>

<html>
<head>
<title>History of Expert System</title>
<!-- #include file="c:\Inetpub\wwwroot\HistoryOfAI\Header.aspx"-->
</head>

<script language="C#" runat="server">
  protected void Page_Load ( object src, EventArgs e ) {
    // specify the data source
    OleDbConnection connGlobal = new OleDbConnection (
      "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
      Server.MapPath ( "~/database/AI.mdb" ) );

    // define the command query
    OleDbCommand myCmd = new OleDbCommand ( "select E_YEAR, E_DESC
from Event where F_ID = 3", connGlobal );

    // open the connection
    connGlobal.Open ( );

    // bind the repeater to the reader
    myRepeater.DataSource = myCmd.ExecuteReader (
CommandBehavior.CloseConnection );
    myRepeater.DataBind ( );

    // close the connection
    connGlobal.Close ( );
  }
</script>

<body bgcolor="#FFFFCC">
<P align="center"><img src="Grapic/cat009.gif"></P>
<br>
<hr size=3 width=90%>

<div class="header"><h3>Expert System</h3></div>

<p align="center">
```

```
<asp:repeater id="myRepeater" runat="server">

 <headertemplate>
   <table width="92%" cellspacing=1 cellpadding=3 style="font:8pt verdana">
   <tr style="background:brown; color:snow; height:15pt">
     <th>Event Year</th>
     <th>Event Discription</th>
    </tr>
 </headertemplate>

 <itemtemplate>
   <tr>
     <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_YEAR" ] %></td>
     <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_DESC" ] %></td>
   </tr>
 </itemtemplate>

 <alternatingitemtemplate>
   <tr style="background-color:beige">
     <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_YEAR" ] %></td>
     <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_DESC" ] %></td>
 </alternatingitemtemplate>

 <footertemplate>
   </table>
 </footertemplate>

</asp:repeater>

</p>

<br>
<hr size=2 width=90%>
<br><br>

<p align="center"><a href="ExpertSystem.aspx"><img
src="Grapic/back_md_wht1.gif" border="0"></a></p>

</body>
</html>
```

Figure 5.4: Sample Code Of HistoryExpertSystem.aspx

This file will display the most recently history is about AI (>= year 2000). The data are displayed using data repeater. E_YEAR>=2001 is to get the result where all the history displayed is more than 2000.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>

<html>
<head>
<title>Recent Event</title>
<!-- #include file="F:\Inetpub\wwwroot\HistoryOfAI\Header.aspx"-->
</head>

<script language="C#" runat="server">
  protected void Page_Load ( object src, EventArgs e ) {
    // specify the data source
    OleDbConnection connGlobal = new OleDbConnection (
      "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
      Server.MapPath ( "~/database/AI.mdb" ) );

    // define the command query
    OleDbCommand myCmd = new OleDbCommand ( "select e.E_YEAR, e.E_DESC,
f.F_NAME from EVENT e, FIELD f where f.F_ID = e.F_ID and E_YEAR >= 2001 "
connGlobal );

    // open the connection
    connGlobal.Open ( );

    // bind the repeater to the reader
    myRepeater.DataSource = myCmd.ExecuteReader (
CommandBehavior.CloseConnection );
    myRepeater.DataBind ( );

    // close the connection
    connGlobal.Close ( );
  }
</script>

<body bgcolor="#FFFF99">
<P align="center"><img src="Grapic/dog0018.gif"></P>
<hr size=3 width=90%>
```

```
<p><strong><font color="#3300ff" size="6" face="cooper black">Recent
Event</font></strong></p>

<p align="center">

<asp:repeater id="myRepeater" runat="server">

  <headertemplate>
    <table width="92%" cellspacing=1 cellpadding=3 style="font:8pt verdana">
    <tr style="background:brown; color:snow; height:15pt">
      <th>Event Year</th>
      <th>Event Discription</th>
                <th>Field</th>
    </tr>
  </headertemplate>

  <itemtemplate>
    <tr>
      <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_YEAR" ] %></td>
      <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_DESC" ] %></td>
                <td><%# ( ( IDataRecord ) Container.DataItem ) [ "F_NAME"
%></td>
    </tr>
  </itemtemplate>

  <alternatingitemtemplate>
    <tr style="background-color:beige">
      <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_YEAR" ] %></td>
      <td><%# ( ( IDataRecord ) Container.DataItem ) [ "E_DESC" ] %></td>
                <td><%# ( ( IDataRecord ) Container.DataItem ) [ "F_NAME"
%></td>
  </alternatingitemtemplate>

  <footertemplate>
    </table>
  </footertemplate>

</asp:repeater>

... ...
```

Figure 5.5: Sample Code Of RecentEvent.aspx

The result.aspx displays the result of the search page. If the results are found in both event and researcher tables, result page will display both the table. If only result only found in event table, only the result in event table will be display and vice versa.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>
<%@ Page Language="C#" Debug="true" ContentType="text/html"
ResponseEncoding="iso-8859-1" %>

<html>
<head>
<title>Search Result</title>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
</head>

<body background="Image/background3.gif" bgproperties="fixed">
<P align="center"><img src="Grapic/dog0018.gif"></P>
<br>
<hr size=3 width=90%>
<div class="header"><h3>Search Result</h3></div>

<p align="left">
<table cellpadding=10>
<%
OleDbConnection connGlobal = new OleDbConnection (
        "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
        Server.MapPath ( "~/database/AI.mdb" ) + "; Persist Security Info=False" );

// define the command query
string cmd1 = "SELECT E_YEAR, E_DESC FROM EVENT WHERE E_DESC LIKE
'%" + Request.Form["E_DESC"] +
                        "%'";
string cmd2 = "select R_NAME, R_DOB, R_MOB, R_YOB, R_DOP, R_MOP,
R_YOP, R_NATIONALITY, R_DESC from RESEARCHER where " +
                        " R_NAME like '%" + Request.Form["E_DESC"] + "%'
OR" +
                        "       R_NATIONALITY       like       '%"       +
Request.Form["E_DESC"] + "%' OR" +
                        " R_DESC like '%" + Request.Form["E_DESC"] + "%'";
OleDbCommand myCmd1 = new OleDbCommand (cmd1, connGlobal);

// open the connection
connGlobal.Open ( );
```

```
OleDbDataReader myReader1 = myCmd1.ExecuteReader();

// Always call Read before accessing data.
string born,died;
string[] months = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sept",
"Oct", "Nov", "Dec"};
int totalEventRec = 0;
int totalResearcherRec = 0;
while (myReader1.Read())
{
        if(totalEventRec == 0)
        {
                Response.Write(
                        "<b>Event(s) Found:</b><table width='100%' cellspacing=1
cellpadding=3 style='font:8pt verdana'>" +
                        "<tr style='background:brown; color:snow; height:15pt'>" +
                                "<th width='15%'>Year</th>" +
                                "<th width='55%'>Event Description</th></tr>");
        }
        Response.Write(
    "<tr><td valign='top'>" + myReader1.GetInt32(0).ToString() + "</td>" +
    "<td valign='top'>" + myReader1.GetString(1) + "</td></tr>");
        totalEventRec++;
}
if(totalEventRec!=0)
{
        Response.Write("</table>");
}
myReader1.Close();


... ....

</body>
</html>
```

Figure 5.6: Sample Code Of Result.aspx

The AddField.aspx is for the administrator to add the new record about field into the database.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>

<html>
<head>
<title>Add Field</title>

... ...

<script language="C#" runat="server">
protected void Page_Load ( object src, EventArgs e ) {
  if ( !IsPostBack ) {
    addPanel.Visible = true;
    postPanel.Visible = false;
  }
  else {
    addPanel.Visible = false;
    postPanel.Visible = true;
  }
}

public void addField ( object src, EventArgs e ) {
  if ( Page.IsValid ) {
    OleDbCommand myCmd = new OleDbCommand ( );
    myCmd.Connection = new OleDbConnection (
      "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
        "F:\\Inetpub\\wwwroot\\HistoryOfAI\\database\\AI.MDB" + "; Persist Security
Info=False" );

    // define insert command parameters
    myCmd.CommandText = "INSERT INTO FIELD ( F_NAME, F_DESC ) VALUES
( ?, ?)";

    myCmd.Parameters.Add ( "Name", OleDbType.VarChar ).Value =
nameField.Value;
    myCmd.Parameters.Add ( "Desc", OleDbType.VarChar ).Value = descField.Value;
    myCmd.Connection.Open ( );
    // do insert
    myCmd.ExecuteNonQuery ( );

    // used for getting new record ID
```

```
myCmd.CommandText = "SELECT MAX(F_ID) FROM FIELD";
        OleDbDataReader myReader = myCmd.ExecuteReader (
CommandBehavior.SingleRow );
        int id=0;
        if(myReader.Read())
            id = myReader.GetInt32(0);
        myReader.Close();
    // fetch new record into reader
    myCmd.CommandText = "SELECT F_ID, F_NAME, F_DESC FROM FIELD
WHERE F_ID=" + id.ToString();
    postDetails.DataSource = myCmd.ExecuteReader ( CommandBehavior.SingleRow
);
    postDetails.DataBind ( );
    myCmd.Connection.Close ( );
  }
}
</script>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1"></head>

<body bgcolor="#FFFFCC">
<div class="header"><h3>Adding a New Field</h3></div>

<hr size=1 width=90%>

<br>

<center>

<asp:panel id="addPanel" runat="server" BackColor="#33FFFF">
  <h5>You can use this form to add a new record.</h5>
  <form runat="server">

  <table width="85%" cellspacing=1 cellpadding=3 border=0>
  <col width=40% align="right">
  <col width=60%>
  <tr>
    <td>Field Name:</td>
    <td><input id="nameField" runat="server">
      <asp:requiredfieldvalidator runat="server"
        controltovalidate="nameField"
        errormessage="Field Name cannot be blank. "
        display="none" /></td></tr>
  <tr>
    <td>Description:</td>
    <td><textarea id="descField" rows=5 cols=35 runat="server" /></td></tr>
```

```
  </table>
  <p>
  <input type=submit value="Add New" runat="server" onServerClick="addField">
  <input type=reset value="Cancel" onClick="self.location.replace ( 'UpdateField.aspx'
) "></p>

  <asp:validationsummary runat="server" displaymode="SingleParagraph"
    showmessagebox=true showsummary=false />

  </form>
</asp:panel>

<asp:panel id="postPanel" runat="server">
  <h5>This record has been added.</h5>

  <asp:datalist id="postDetails" width="85%" runat="server">

    <itemtemplate>
      <table  align="center" id="posted" width=100% cellspacing=1 cellpadding=5
border=0>
      <col width=35% align="right">
      <tr>
        <td valign='top'>Field Name:</td>
        <td valign='top'><b><%# ( ( IDataRecord ) Container.DataItem ) [ "F_NAME"
] %></b></td></tr>
      <tr>
        <td valign='top'>Description:</td>
        <td valign='top'><b><%# ( ( IDataRecord ) Container.DataItem ) [ "F_DESC" ]
%></b></td></tr>
      </table>
    </itemtemplate>
  </asp:datalist>
  <P align="center"><a href="UpdateField.aspx"><img
src="../Grapic/back_md_wht1.gif" border="0"></a></p>
</asp:panel>
</center>
<br>
<hr size=1 width=90%>
</body>
</html>
```

Figure 5.7: Sample Code Of AddField.aspx

The UpdateField.aspx is for the administrator to update the existing record of the field in the database. The data are all display inside a table.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>
<%@ Page Language="C#" Debug="true" %>

<html>
<head>
<title>Update Field</title>

... ...

<script language="C#" runat="server">
OleDbConnection myConn = new OleDbConnection (
      "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
        "F:\\Inetpub\\wwwroot\\HistoryOfAI\\database\\AI.MDB" + "; Persist Security
Info=False" );
OleDbCommand myCmd = new OleDbCommand ( );
string id;

protected void Page_Load ( object src, EventArgs e ) {
  myCmd.Connection = myConn;
  id = Request.QueryString [ "id" ] ;
  if ( !IsPostBack ) {
    if ( id == null ) {
      bindGrid ( );
      loadPanel.Visible = true;
      editPanel.Visible = false;
      postPanel.Visible = false;
    } else {
      getField ( Request.QueryString [ "id" ] );
      loadPanel.Visible = false;
      editPanel.Visible = true;
      postPanel.Visible = false;
    }
  } else {
    loadPanel.Visible = false;
    editPanel.Visible = false;
    postPanel.Visible = true;
  }
}
```

```
public void bindGrid ( ) {
  myCmd.CommandText = "SELECT F_ID, F_NAME, F_DESC FROM FIELD
ORDER BY F_ID";
  myConn.Open ();
  myGrid.DataSource = myCmd.ExecuteReader ( CommandBehavior.CloseConnection
);
  myGrid.DataBind ( );
}

public void getField ( String id ) {
  OleDbDataAdapter myAdapter = new OleDbDataAdapter (
    "select F_ID, F_NAME, F_DESC from FIELD where F_ID=" + id, myConn );
  DataTable fieldDetails = new DataTable ( );
  myAdapter.Fill ( fieldDetails );
  DataRowView myRow = fieldDetails.DefaultView [ 0 ] ;

  fieldName.Value = myRow [ "F_NAME" ].ToString ( );
  fieldDesc.Value = myRow [ "F_DESC" ].ToString ( );
  editPanel.DataBind ( );
}

public void updateMessage ( object src, EventArgs e ) {
  if ( Page.IsValid ) {
    myCmd.CommandText = "UPDATE FIELD SET F_NAME=?, F_DESC=? where
F_ID=" + id;

    myCmd.Parameters.Add ( "Name", OleDbType.Char ).Value = fieldName.Value;
    myCmd.Parameters.Add ( "Desc", OleDbType.Char ).Value = fieldDesc.Value;

    myConn.Open ( );
    myCmd.ExecuteNonQuery ( );
    myConn.Close ( );
  }
  bindPostPanel ( id );
}

public void bindPostPanel ( String id ) {
  myCmd.CommandText = "SELECT * FROM FIELD WHERE F_ID=" + id;
  myConn.Open ( );
  listDetails.DataSource = myCmd.ExecuteReader ( CommandBehavior.SingleRow );
  listDetails.DataBind ( );
  myConn.Dispose ( );
}
</script>
</head>
```

```
... ...

  <asp:datagrid id="myGrid" runat="server"
    width=90% cellpadding=5 gridlines="vertical"
    bordercolor="black" borderwidth=1
    font-size="8pt"
    backcolor="ghostwhite"
    alternatingitemstyle-backcolor="lightgray"
    autogeneratecolumns=false>

    <headerstyle backcolor="darkslategray"
      forecolor="khaki" height=25 font-bold />

... ...
  <p>
  <a href="javascript:history.go ( -1 )"><img border="0" alt="Back to Edit View"
onmouseover="status='Back to Edit View'" img src="Image/button25.gif""></a>
  <a href="UpdateField.aspx"><img border="0" alt="Select Another Record"
onmouseover="status='Select Another Record'" img src="Image/button24.gif""></a>
  </p>

</asp:panel>

</center>
<br>
<hr size=1 width=90%>
</body>
</html>
```

Figure 5.8: Sample Code Of UpdateField.aspx

The DeleteField.aspx is to delete the record of field from the database. The data are all display inside a table.

```
<%@ Import Namespace="System.Data" %>
<%@ Import Namespace="System.Data.OleDb" %>

<html>
<head>
<title>Remove Field</title>

.....

<script language="C#" runat="server">
OleDbConnection myConn = new OleDbConnection (
      "Provider=Microsoft.Jet.OleDb.4.0; Data Source=" +
        "F:\\Inetpub\\wwwroot\\HistoryOfAI\\database\\AI.MDB" + "; Persist Security
Info=False" );
OleDbCommand myCmd = new OleDbCommand ( );
String id;

protected void Page_Load ( object src, EventArgs e ) {
  myCmd.Connection = myConn;
  id = Request.QueryString [ "id" ] ;
  if ( id == null || IsPostBack ) {
    if ( id == null ) bindGrid ( );
    loadPanel.Visible = true;
    removePanel.Visible = false;
  }
  else {
    getField( id );
    loadPanel.Visible = false;
    removePanel.Visible = true;
  }
}

public void bindGrid ( ) {
  myCmd.CommandText = "SELECT F_ID, F_NAME, F_DESC FROM FIELD
ORDER BY F_ID";
  myConn.Open ();
  myGrid.DataSource = myCmd.ExecuteReader ( CommandBehavior.CloseConnection
);
  myGrid.DataBind ( );
  myConn.Close ( );
}
```

```
public void getField ( String id ) {
  myCmd.CommandText = "SELECT F_ID, F_NAME, F_DESC FROM FIELD
WHERE F_ID=" + id;
  myConn.Open ();
  listDetails.DataSource = myCmd.ExecuteReader ( CommandBehavior.SingleRow );
  listDetails.DataBind ( );
  myConn.Close ( );
}

public void removeField ( object src, EventArgs e ) {
  myCmd.CommandText = "DELETE from FIELD where F_ID=" + id;
  myConn.Open ( );
  myCmd.ExecuteNonQuery ( );
  myConn.Close ( );
  bindGrid ( );
}
</script>
</head>

.....

  <p align="center"><a href="#top"><img alt="Back to top"
onmouseover="status='Back to top of page'" img src="../Grapic/bearblue.gif"
border="0"></a></p>
        <p align="right"><a href="AdminMain.aspx"><img
src="../Grapic/back_home_md_wht.gif" border="0"></a></p>

</asp:panel>

<asp:panel id="removePanel" runat="server">

  <h5>To remove this field, press Delete.</h5>

  <form runat="server">

    <asp:datalist id="listDetails" width="85%" runat="server">

      <itemtemplate>
        <table id="remove" width=100% cellspacing=1 cellpadding=5 border=0>
        <col width=35% align="right">
        <tr>
          <td valign='top'>Field Name:</td>
          <td valign='top'><b><%# ( ( IDataRecord ) Container.DataItem ) [
"F_NAME" ] %></b></td></tr>
        <tr>
```

```
<td valign='top'>Description:</td>
        <td valign='top'><b><%# ( ( IDataRecord ) Container.DataItem ) [ "F_DESC"
] %></b></td></tr>
      </table>
    </itemtemplate>

  </asp:datalist>

  <p>
  <input type=submit value="Delete" runat="server"
    onServerClick="removeField">
  <input type=reset value="Cancel"
    onClick="self.location.replace ( 'DeleteField.aspx' ) "></p>

  </center>

 </form>
</asp:panel>
</center>
<br>
<hr size=1 width=90%>
</body>
</html>
```

Figure 5.9: Sample Code Of DeleteField.aspx

6.5 Problem Encountered and Solution

6.5.1 Lack of knowledge in C# and ASP.NET

A lot of resources are available in the Internet about C# and ASP.NET. Most of them could be found explained in the Internet and this has become one of the solutions for this particular problem encountered. Besides than that, course syllabus for information technology is not including the web-based system development as well as the C# programming. The misunderstanding will be happen some time.

The knowledge on irregularities and unorganized coding style and methods is most serious problems. There have no standard style of code is used with the reason that misunderstanding different sample codes in the reference books.

Seeking friend's and lecturer's help and trying to understand the coding using longer time is the ways to overcome these problems. Internet also provides a lot of resources for the solutions when the problems were found.

5.5 Problem Encountered and Solution

5.5.1 Lack of knowledge in C# and ASP.NET

A lot of resources are available in the Internet about C# and ASP.NET. Most of them could be found explained in the Internet and this has become one of the solutions for this particular problem encountered. Besides than that, course syllabus for information technology is not including the web-based system development as well as the C# programming. The misunderstanding will be happen some time.

The knowledge on irregularities and unorganized coding style and methods is most serious problems. There have no standard style of code is used with the reason that misunderstanding different sample codes in the reference books.

Seeking friend's and lecturer's help and trying to understand the coding using longer time is the ways to overcome these problems. Internet also provides a lot of resources for the solutions when the problems were found.

Chapter 6 – Testing and Evaluation

6.1 System Testing

Software testing of the website is an extremely creative and intellectually challenging task, which considered as a part of software engineering. Software testing for the website involves operation of a system or application under controlled conditions and evaluating the results. Normal and abnormal conditions are included in the controlled conditions. Software testing help should intentionally attempt to make things go wrong to detect the incorrect things happen and the necessary things do not work.

Below are the objectives of software testing for the system:

- To minimum amount of time and effort on rework
- To increased the quality by less on "bug fixing"
- To make sure the requirements and specification of the website have been meet

6.2 Testing Principles

Testing principles should be clearly understood to make sure the software testing will be done smoothly and successfully. Bellow are some of the principles that must be followed to have the creative element of test design and execution rivals any of the preceding development steps.

The principles are: -

- Testing should not be planned under the tacit assumption that no errors will be found.
- Test for invalid and unexpected input conditions as well as valid conditions.
- Keep system static during test.
- Plan the cases of testing and make document of it and the testing result

6.3 Testing Strategies

There have a lot of strategies could be used in the software testing of the system. The testing strategies was divided into 3 main tests which are, Unit Testing, Integration Testing and System Testing

6.3.1 Unit testing

Unit test is the process to test the individual and the smallest component to ensure that function work properly. Each component is tested independently. Unit test is performed concurrently with the development process to test particular functions or code modules. Detailed knowledge of the internal program design and code is required in unit testing. Unit testing is easily done as the application has a well-designed architecture with tight code; may require developing test driver modules or test harnesses.

There have 5 procedures in unit testing inside this system: -

- Review the design documentation which was done in the previous phases

- Establish test categories such as path execution, I/O checks, interface data correctness, bounds, error handling
- Develop tests for each category
- Specify expect results
- Define "driver" or "stub" software required for tests.

Several techniques were used during the process of performing unit testing.

a) Code Review

Codes are reviewed line by line. The purpose of codes review is to discover any syntax error as well as semantic error. Discovered errors are corrected immediately.

b) Boundary Value Analysis

Input specifications and output specifications, which will generate a small set of test data for detecting potential faults, were tested.

c) Error Handling Paths

When the errors occurred, ensure the specific program execute the recovering process.

d) Executing Program Paths

To check all the link and buttons are well functioning and bring user to destination page. The alignment of image and the *onmouseover* function are also tested.

6.3.2 Integration Testing

Parts of an application such as code modules, individual applications, and client and server applications on a network were combined to determine if they function together correctly. This type of testing is especially important to client/server and distributed systems.

Integration testing provides systematic approaches for assembling the software in an incremental fashion. There are generally two approaches, which are top down integration and bottom up integration.

- The approach that is used in integration testing for this system is bottom up integration. Bottom up testing involves integrating and testing the modules at the lower levels in the hierarchy, and then working up the hierarchy of modules until the final module is tested. This approach does not require the architectural design of the system to be completed so that it can start at an early stage in the development process.

6.3.3 System Testing

System testing cover all combined parts of a system by integrates the sub-systems to make the entire system. System testing is black-box type testing which not based on any knowledge of internal design or code that is based on overall requirements specifications. The purpose of the system testing is to find errors that result from unanticipated interactions between sub-systems. System testing also fully executes the system to uncover its limitation and measure its capability. System testing also makes sure that the system meets its functional and non-functional requirement.

There is several type of system testing, which included: -

a) Function Testing

This testing focuses on the functionalities of the system. Black-box type testing geared to functional requirements of an application. This testing has high probabilities in detecting the faults and this are increased the quality of the system.

In the Function Testing, Expected actions and outputs are known and anticipated. All the valid and invalid input data types are used to test. Stopping criteria is included in the Function Testing of the system.

This testing is important to ensure the system have the high ability to perform the function, which mentioned in Chapter 4.

b) Performance Testing

Performance testing used interchangeably with stress and load testing. Ideally performance testing is performed to compare the integrated modules with the non-functional system requirements.

In this testing, the system response time degrades or fails is tested. Performance testing is useful to describe such tests as system functional testing while under unusually heavy loads, heavy repetition of certain actions or inputs, input of large numerical values, and large complex queries to a database system.

c) Acceptance Testing

Acceptance Testing is the final testing based on specifications of the end-user or customer, or based on use by end-users/customers over some limited period of time. Users of the website lead the acceptance testing and define their own time data sets that used as test cases. Acceptance testing could determine that if the website could meet the performance expectation.

The different of the acceptance testing and the system integration testing is only the tester. The testers here are the users of the website where the students in the University of Malaya did the acceptance testing especially the students in the Faculty Computer Science and Information Technologies. The acceptance testing helps to ensure that the system meets all the requirements and specification and make sure that it is ready as a production.

6.4 User Evaluation

6.4.1 Objective

Upon the application testing, the users of the website were testing the system and also evaluate the system to assess the effectiveness and the general outlook of the systems. This evaluation help to gauge the accuracy of the system and make sure the requirements of the users are fulfilling. The students in the University of Malaya carry out this testing and evaluation.

6.4.2 Summary on the Results

a) Interface Design

In the summary, the interface design is simple and nice for the navigation. The interface could help users who had never been to the site easily figure out where to go by click on the menu button. The banner of in the web site attracts most of the users.

The user noted that the download times of the web site are fast and this help the user saving their time from waiting patiently for the page to load. The image and the traffic in the Internet is the reason that the page loading faster.

The cascading styles in the web site is acceptance by most of the users and only a few user states that too normal of the text styles. The color of the table also attract interested from the user. Conclusion, user quite satisfied with the interface design of the web site.

b) Functions of the System

The summary shows that the website is displayed quite a lot data. Besides that, the administrator can add, edit and delete the data online at anytime and anywhere.

c) Suggestions

There are a lots of suggestion are given by the users. In the summary, lecturers suggested that add more information about the history. The information should be more hypermedia-based.

6.5 Error Handling and Debugging

Error handling helps the developers to have a clearer development. The error handling makes sure that the program is more robust and fault tolerant. There is some ability on recovering the infrequent errors from suffering the consequences in the error handling.

The debugging is useful in the process of finding and collecting the errors or bugs. There are a lot of the debugging tools in the cyber market now, which is including Toggle Breakpoint, Step Into and so on. The locals and immediate windows are used when debugging the errors.

Chapter 7 – System Evaluation

7.1 Strength and Significance of the System

Below is the strength and significance of the system: -

a) **Dropdown menu**

The dropdown menu is easy to navigate. Besides, it can save the space of the website and make the website looks nicer and more attractive. Besides that, the dropdown menu is very colourful and is placed in most of the pages. As a result, it is easier for the user to navigate.

b) **Data**

The data are displayed in a systematic way so that the user will be attracted. Besides, it is in tidy form so that the user will feel comfortable to look through the website. It helps the user to save time..

c) **Simple and user friendly interface**

The color / interface of the website are attractive. Besides, it is user friendly. The user won't feel boring navigate through the website. The graphics enhance the website make it more attractive and nicer.