Sekolah Siswazah
(Graduate School)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certification of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certify that)

MOHD KHALID BIN AWANG

calon untuk Ijazah
(candidate for the degree of) Sarjana Sains (Teknologi Maklumat)

telah mengemukakan kertas projek yang bertajuk
(has presented his/her project paper of the following title)

AN EVALUATION OF ARTIFICIAL NEURAL NETWORK
IN PREDICTING THE PRESENCE OF HEART DISEASE

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project paper)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan, dan meliputi bidang ilmu dengan memuaskan.
(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).

Nama Penyelia
(Name of Supervisor): Puan Fadzilah Siraj

Tandatangan
(Signature) :

Tarikh
(Date) : 27 September 2000

AN EVALUATION OF ARTIFICIAL NEURAL NETWORK IN PREDICTING THE PRESENCE OF HEART DISEASE

A project submitted to the Graduate School in partial
fulfilment of the requirements for the degree
Master of Science (Information Technology),
Universiti Utara Malaysia

by
Mohd Khalid Bin Awang

PERMISSION TO USE

In presenting this project in partial fulfilment of the requirements for a postgraduate degree from the Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for copying of this project in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor(s) or, in their absence, by the Dean of the Graduate School. It is understood that any copying or publication or use of this project or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any material from my project paper.

Requests for permission to copy or to make other use of material in this project in whole or in part should be addressed to:

Dean of Graduate School
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman

ABSTRACT (BAHASA MALAYSIA)

Projek ini bertujuan untuk menilai keupayaan model rangkaian neural dalam meramal penyakit jantung terutama kewujudan 'angina' pada pesakit yang telah dikenalpasti menghidap 'myocardial infarction'. Ramalan tentang kewujudan 'angina' adalah penting dalam menentukan kaedah rawatan yang bersesuaian untuk pesakit tersebut. Tambahan pula, diagnostik dan pengurusan 'angina' adalah penting untuk mencegah pengulangan serangan 'myocardial infarction'. Pembangunan aplikasi ini melibatkan tiga fasa utama. Fasa pertama ialah pembangunan Sistem Pengurusan Maklumat Myocardial Infarction (MIMIS) yang bertujuan untuk mengumpul dan mengurus maklumat pesakit. Kemudian diikuti oleh fasa kedua iaitu pembangunan Simulasi Rangkaian Neural (NNS) dengan menggunakan model perambatan balik untuk melatih dan menguji rangkaian. Fasa terakhir ialah pembangunan Sistem Ramalan (PS) untuk membuat ramalan terhadap pesakit yang baru. Kesemua sistem tersebut dibangunkan dengan menggunakan perisian Microsoft Visual Basics. Data untuk sessi latihan dan ujian diperolehi daripada Hospital Besar Alor Setar, Kedah. Model rangkaian neural terbaik yang dihasilkan mampu mencapai ketepatan ramalan sehingga 88.89 peratus. Kaj ian ini bukan sahaja membuktikan kemampuan rangkaian neural dalam mendiagnosis penyakit, malahan juga berjaya mengabungkan rangkaian neural dengan sistem pengurusan maklumat. Sebagai projek perintis, aplikasi ini dapat digunakan sebagai model untuk membangunkan sistem sokongan keputusan perubatan, terutama didalam mengdiagnosis penyakit jantung.

ABSTRACT (ENGLISH)

The purpose of this study is to evaluate the application of artificial neural network in predicting the presence of heart disease, particularly the angina in patients that already diagnosed with myocardial infarction. The prediction and detection of angina is important in determining the most appropriate form of treatment for these patients. Furthermore, diagnosis and management of angina is important since it can lead to the recurrent of myocardial infarction. The development of the application involves three main phases. The first phase is the development of Myocardial Infarction Management Information System (MIMIS) for data collection and management. Then followed by the second phase, which is the development of Neural Network Simulator (NNS) using back propagation for network training and testing. The final phase is the development of Prediction System (PS) for prediction on new patient's data. All systems had been developed using Microsoft's Visual Basics. The data used to train and test the network was provided by Alor Setar General Hospital, Kedah. The best network model produced prediction accuracy of 88.89 percents. Apart from proving the ability of neural network technology in medical diagnosis, this study also shown how the neural network could be integrated into a management information system as a prediction tools. As the pilot project, the application developed could be used as the starting point in building a medical decision support system, particularly in diagnosing the heart disease.

ACKNOWLEDGEMENTS

The development of this project has been a long journey. Throughout this journey, I was fortunate to have had the help and contributions of my supervisors, Puan Fadzilah Siraj. I would like to extend my thanks to my beloved wife, Azizah, my daughters, Nur Hadiyah and Nur Hazwani for the courage and understanding. This project would not been possible without their encouragement, support and guidance.

I would also like to thank the Alor Setar General Hospital staffs, especially Dr Nizar for supplying the data of myocardial infarction. The data and advices have been tremendously useful in the development of the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

Worldwide, cardiovascular disorders and heart disease are considered as the number one killer. Although the death rates from the diseases are now on the decrease as a result of changes in lifestyle, it is still a major cause of death and disability in both developing and developed countries. Cardiovascular disorders claimed 953,110 lives in the United States in 1997 and it contributed 41.2 percent of all deaths or 1 of every 2.4 deaths. A cardiovascular disorder was also as a primary or contributing cause on over 1,406,000 death certificates (American Heart Association, 2000). People in the developing country including Malaysian are also exposed to the same risk. According to our Ministry of Health (1997), heart disease and cardiovascular disorders contributed up to 20.26 percent of all total deaths in Malaysia or equivalent to 8,915 of death yearly.

The main function of the cardiovascular system is to supply vital oxygen as well as nutrients to all body cells and tissues will die within minutes without this supply (Calnan, 1991). A cardiovascular disorder is a medical terminology that describes all kind of diseases that related to our heart and circulatory system. The coronary heart disease (CHD) is one of the most dangerous heart problems, which occurs when the heart muscle receives insufficient oxygen because the coronary arteries fail to maintain a sufficient supply of blood (Calnan, 1991). According to Open University (1985), coronary heart disease threatening people in three main ways by producing:

1) Angina, which also known as chest pain. It can cause considerable debilitation. It occurs when vessels that carry blood to the heart become narrowed and blocked due to atherosclerosis.
2) Myocardial infarction and also called heart attack. It is a condition where a part of the heart muscle is permanently damaged. The coronary artery becomes completely blocked off and the cells in part of the heart muscle are permanently injured.
3) Sudden death or always referred as heart failure. This condition occurs when the heart muscle suddenly stopping and it is usually due to thrombosis on a plaque.

1.1 Problem Statement

The statistics presented in the previous section reveal that heart disease and cardiovascular disorders are one of the world's most important causes of mortality, so improvement of diagnosis procedure would be very vital. Even though there is still considerable uncertainty about the cause of heart disease that make the diagnosis more difficult, it also reveals that coronary anatomy can be predicted from finding and the initial encounter such smoking habits and cholesterols level (Calnan, 1991). Proyor et al., (1993) indicated that a lot of life could be saved and at the same time some of the total cost related to the diagnostic testing could be reduced if the initial assessments are reliable and further testing only be given to highly risk patients. There have been previous efforts that use statistical methods to predict the presence of heart disease in patients; however, this method may not be able to identify all the hidden relationship between the various risk factors of the heart diseases that make the result not satisfactory. Diagnosing heart disease is considered as a non-linear problem that shows complex causal relationship between the variables (Calnan, 1991). However, there is new computer paradigm called an artificial neural network, which is suitable for problems of extremely complexity not addressable with our conventional technologies, either by the conventional computer programming or statistical method. Several studies shown that an artificial neural network can be successfully applied in diagnosing heart diseases (Harrison et al., 1991; Baxt, 1991). Therefore, the purpose of this study is to evaluate the application of neural network in predicting the presence of heart disease, particularly the angina in patients. The prediction is mainly used for diagnosis angina in patients that are already diagnosed with myocardial infarction. The diagnosis of angina is important in determining the most appropriate form of treatment for myocardial infarction's patients. Furthermore, diagnosis and management of angina is important for these patients since the angina can lead to the recurrent of myocardial infarction (Calnan, 1991). Nevertheless, the prediction system can also be used for new patients when they are admitted to the hospital with some sort of chest pain.

Figure 1.1 shows the proposed diagnosis system for detecting angina in patients. Patient's information together with the details information about the heart disease or the risk factors of angina is gathered. In a current situation, all patients will go through the exercise stress test (EST) without taking into consideration about the patient's condition. Therefore, all patients whether the low, moderated and high risk will be treated equally.

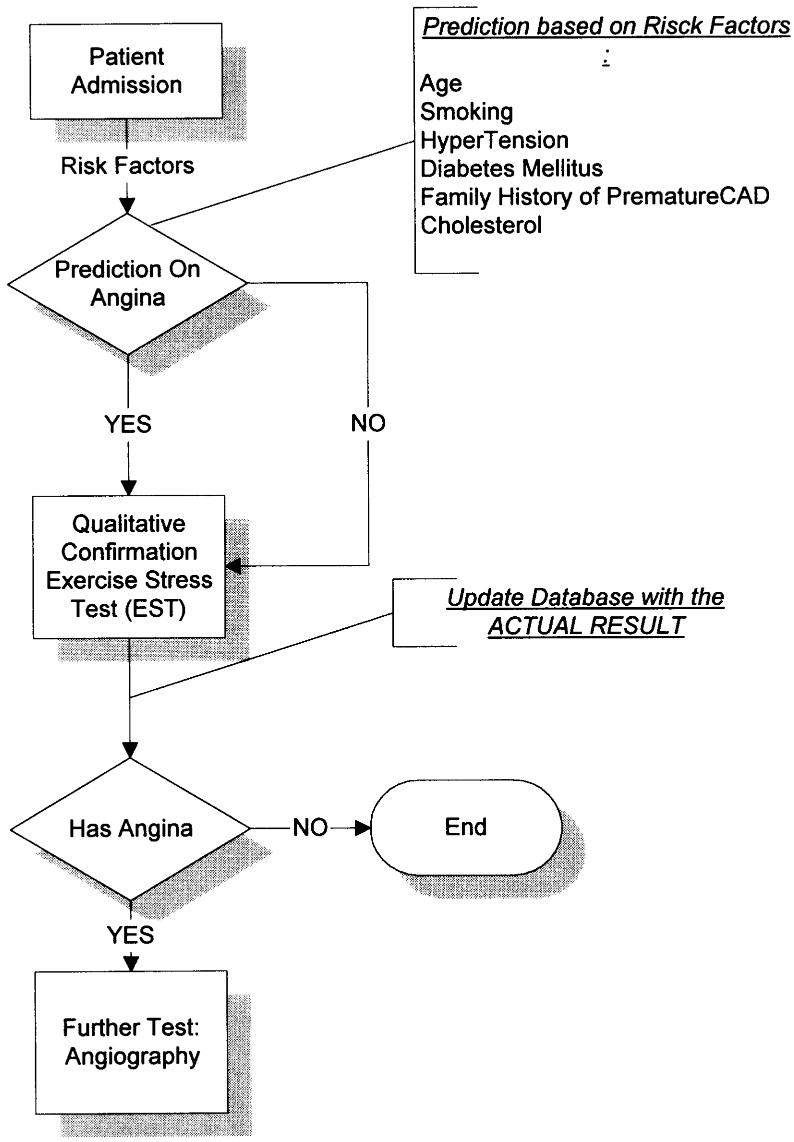


Figure1.1: Proposed Diagnosis System for Detecting Angina in Patients

The new proposed system will make an early prediction on the patient's form, whether or not to suffer from angina. If the patient is predicted to have an angina, then the physician would supervise the EST directly. However, if the patient is predicted for not having an angina, then the EST would be performed with a physician in the immediate distance or only monitored by a certified officer in Basic Life Support as suggested by National Heart Attack Alert Program Working Group (1997). The actual finding will be recorded and the patient's database will be updated for further neural network retraining process. However, if the prediction system performed very well, reliable and consistent, then the EST for the patient who was predicted without having an angina could be ignored. Therefore, some cost could possibly be saved. In addition, physicians can concentrate more on high-risk patients.

1.2 Objectives

The objective of this study is to evaluate the application of neural network in predicting the presence of angina in patients. Specifically, the sub-objectives of the study are as follows:

1. To develop a model for Management Information System for heart diseases (Myocardial Infarction).
2. To identify the risk factors that contributes to angina.
3. To develop a Neural Network Simulator to be used in training and testing.
4. To develop a model for automatic angina prediction system. The system will get the input form the Management Information System interface and then used the knowledge of trained neural network to predict the presence of angina in patients.
5. To identify the most suitable neural network model for predicting the angina.
6. To evaluate the performance of the system in predicting whether the patients are suffering from angina.

1.3 Scope of the Study

This study focuses on the prediction of angina in patients whom already being diagnosed with myocardial infarction. However, the application developed in this study can also be used for diagnosing new patients. The primary data for the study are collected from Alor Setar General Hospital. Some patient's data from UCI Machine Learning database are also used for comparison.

The study is not meant for developing a comprehensive Management Information System (MIS) for heart diseases. However, in this study, an application model that shows the interaction and integration between all main modules for MIS were developed. Initially, data are entered through the interface of Myocardial Infarction Management Information System and then exported into Neural Network Simulator for training and testing. Once the prediction capability of the network is satisfied, then final knowledge will be stored in the weight file. The weights or knowledge from trained neural network are utilized by the Prediction System to make prediction on new patient's data.

All modules are written in Microsoft Visual Basic 5.0 programming language and Microsoft Access 1997 is chosen as the database engine. In addition, Crystal Report is used in developing reports for Myocardial Infarction Management Information System.

1.4 Significance of the Study

The project is significant since it is the first project initiated to predict angina in patients. This is also the first effort in computerization of cardiovascular medical record, particularly the myocardial infarction in Alor Setar General Hospital. Specifically, this study will have few advantages as follows:

1. Helps clinicians in recording, organizing, manipulating and retrieving patient's records.
2. Reduce the time, costs and efforts in diagnosing patient.
3. Help technicians in their early decision-making by giving advise through automatic prediction.
4. As a pilot project in evaluating and implementing the computer information system in Alor Setar General Hospital.
5. The project also serves as an example on how the neural network technology can be successfully utilized in medical domains especially in prediction of disease.

Chapter 2 Neural Network

Generally, human activities involve two basic categories of problems. Firstly, the structured problems which are clearly defined and deterministic; they can be solved by computer with precise logic and clear algorithms. However some problem such as pattern recognition and undersea mining are considered as ill-posed problems (Poggio et al., 1985). These unstructured problems cannot be solved by conventional computational method (Aho et al., 1974). However, a 3 years olds child can solve these unstructured problems such as human face recognition easily. This is the fact that, our mind is the most powerful and perfect computer that ever exists. Therefore, based on this scenario, a lot of efforts have been made to produce a brain-like computer and finally, an artificial neural network is created and it is not more than an imitation of human neural processing system.

Specifically, neural network can be defined as " a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes" (DARPA, 1988). In general, neural network is an attempt to understand brain-like computational systems (Ballard, 1986; Kohonen, 1977) and sometimes referred as computational neuroscience (Amari et al., 1982) and also known as connectionism (North, 1987). McCulloch and Pitts (1943) were among the first pioneers in the neural network research. They developed mathematical models based on the real human neurons. However, the neural network became popular after the development of the back propagation algorithm by Rumelhart et al., (1986). Nowadays, neural networks have been applied in various fields such as finance, marketing, engineering and medical domains.

Most existing neural network models are based on the human neurons. Figure 2.1 and 2.2 shows the comparison between the real and an artificial neural configuration. The understanding of the human neurons has led to the formulation of artificial neurons as a basic building block of neural network model (Shakoff, 1999). The neurons (also known as processing units) are highly connected with one another. The connection is call as weight.

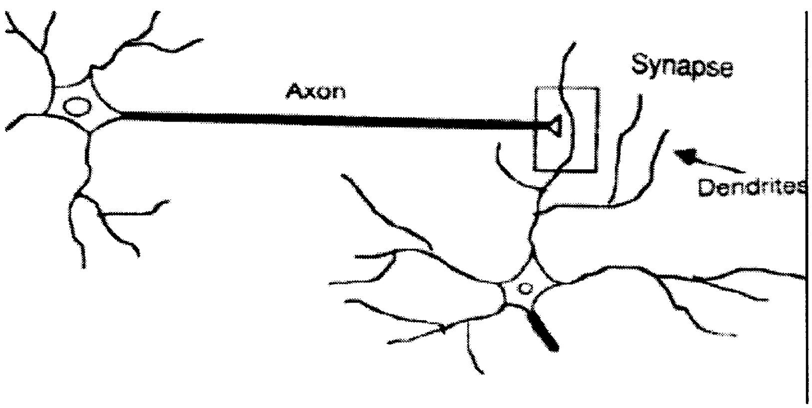


Figure 2.1: Human Neuron

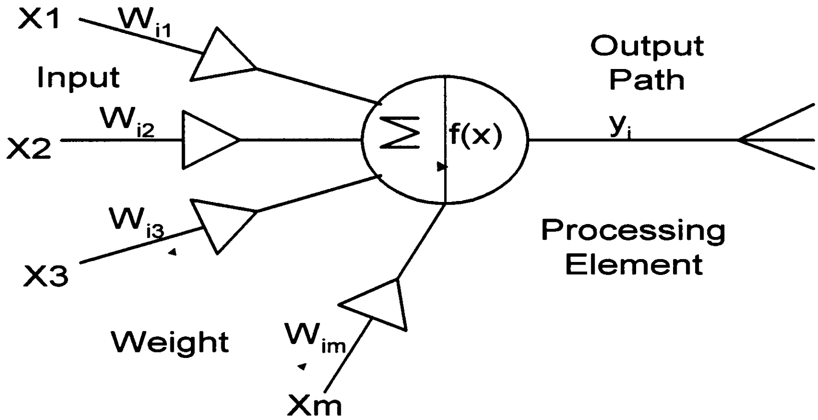


Figure 2.2: Artificial Neuron

As human neuron, each processing unit consists of numbers of connection of input connection like neuron's dendrites. It also has one or numbers of output units as neuron's axon. Each processing unit has an energy potential or known as state of activity, which is the same as neuron's internal potential (Fausett, 1994). In short, the processing unit in an artificial neural network has three main processes, which is receiving inputs, compute activation level based on given inputs and finally spread the computed result or output to other connected units.

2.1 Neural Network Learning Method

Initially, there is no knowledge implemented or programmed to neural network, but it only supply with a learning principle. When neural network is exposed to learning environment by giving them new training patterns, it will use the implanted learning methodology to learn new knowledge (Fausett, 1994). This learning process is carried out based on a variety of learning algorithm. Currently there are three types of learning algorithms that are widely used

(Bigus, 1996). The first is the supervised algorithm that applies student and teacher teaching-learning principle and good in performing classification and prediction problems. In this methodology, neural network is supplied with input and desired output or target value. The second algorithm is the unsupervised, which has no desired output. The network will learn the pattern of training data and find out the relationship between the variables or inputs and it is suitable in clustering purposes. Finally, the reinforcement-learning algorithm requires examples of the problem or case, however the exact answers to the problem are not available.

2.2 Back Propagation Neural Network

The first and the easiest learning rule for artificial neural network was designed by Hebb (1949) which known as Hebb learning rule. In 1998, McClelland and Rumelhart enhanced the Hebb learning rule. Among the first researchers who made big improvement in neural network was Rosenblatt (1959). He developed a large class of single layer network called perceptrons. The perceptron learning rule utilizes an iterative weight adjustment that is more powerful than the firs Hebb rule. The early accomplishment of perceptrons led to the enthusiastic claims of neural network technology. People started to realize the power of neural network. However, in 1969, Minsky and Papert demonstrated that many of the desirable mappings for non-linear problems were unachievable with perceptrons. Due to this discovery, research in neural network once again became less active and known as 'quiet years' (Fausett, 1994).

Neural network reemerges again when the development of back propagation learning algorithm was developed in the mid 1980s. The back propagation network uses a multi-layer perceptrons for which the limitation of the old single-layer neural network can be solved. The algorithm was independently discovered by (Werbos, 1974; Lecun, 1985; Parker, 1985) and (Rumelhart, Hinton and Williams, 1986). Nowadays, most of the neural network applications utilize this algorithm (Tsoukalas and Uhrig, 1997). In particular, the emergence of back propagation network has contributed to the application of neural network in medical domains (Ohno-Machano, 1996).

A back propagation neural network uses a feed forward topology with supervised learning and back propagation learning algorithm. It is also widely known as Multi Layer Perceptron (MLP). Several applications such as NETtalk, which learned to read English aloud, were based on back propagation network (Bigus, 1996).

2.3 Back Propagation Architecture

The basic building block of back propagation network with one hidden layer and biases is illustrated in the following figure.

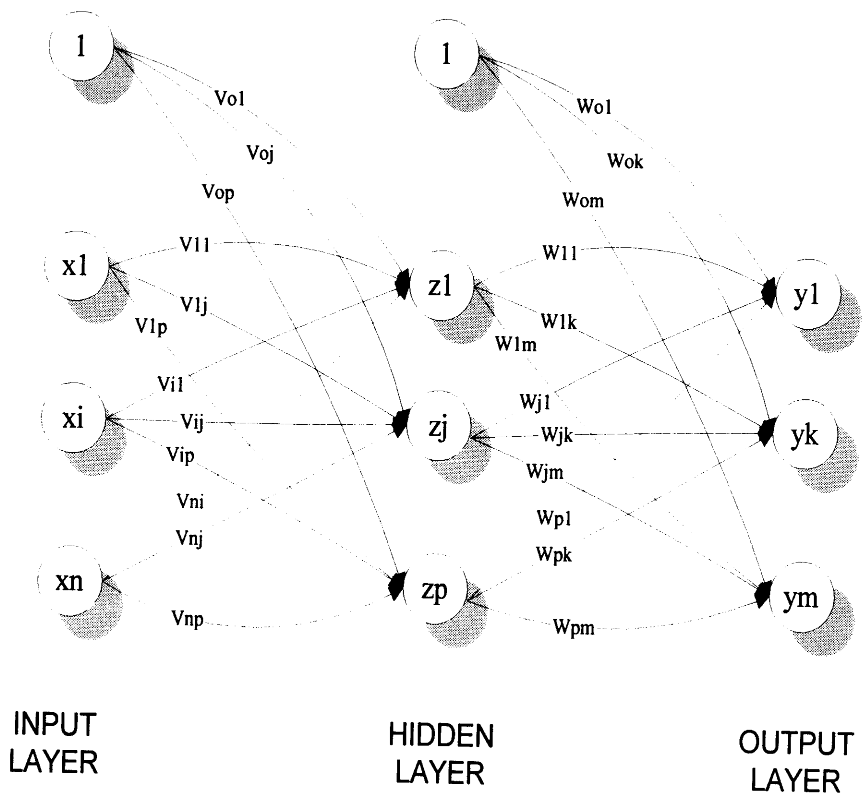


Figure 2.3: Back Propagation Neural Network with One Hidden Layer

2.4 Back Propagation Algorithm

Basically there are 3 stages involved in back propagation network (see Figure 2.4); the feed forward of input training patterns, then followed by calculation of back propagation of the associated error and finally the adjustment of the weights for all layers (Fausett, 1994).

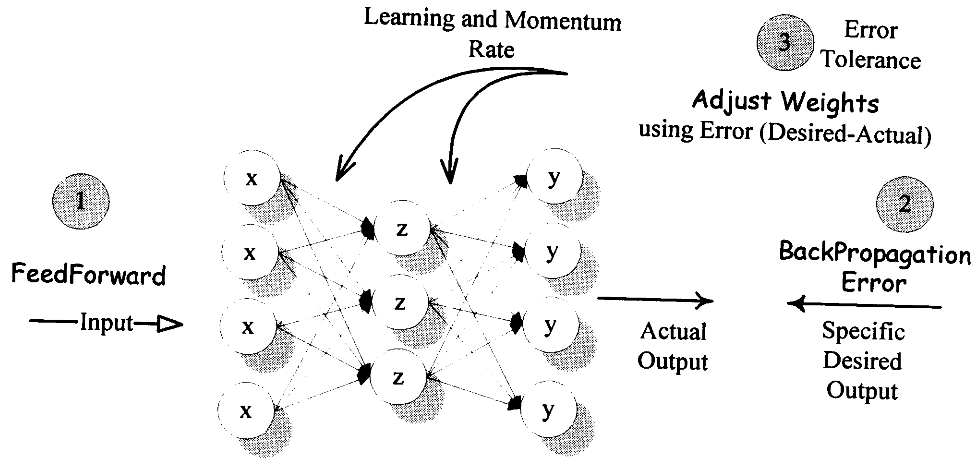


Figure 2.4: Back Propagation phases

Feed Forward Phase

The first stage is the feed forward, where each input node receives an input signal and broadcasts the signal to each node of the upper layer. This signal becomes the input for the hidden nodes. Then each node in the hidden layer will sum up the signals of all the input nodes. Based on the information, the hidden node computes its activation and sends its signal to the upper layer, which is the output node. Then each output node calculates its activation function to generate its output signal. This output signal is the network-calculated output for the given input pattern (Fausett, 1994).

Back propagation Phase

During the back propagation stage, each output node compares its calculated output or computed activation with its actual output or target value to determine the associated error for that pattern with that node. Based on the error, the error correction weight adjustment for the hidden and the output units is computed.

Later, this calculation is used to update the weights between the output and the hidden layers. Similarly, the error correction of weight adjustment for the hidden and the input units is computed too. The weight between the hidden and the input layers will be updated with this factor (Fausett, 1994).

Update Weight Phase

In the weight adjustment stage, the weights for all layers are adjusted simultaneously. The adjustment of weights is based on both factors calculated above.

2.4.1 Back Propagation Training Algorithm

The fundamental concept of the algorithm and related mathematical calculation formula for the back propagation training process is described as follows:

Step 1.0: Initialize weights

(Set to a small random values between 0 to 1 or –0.5 to 0.5)

Step 2.0: While Stopping Condition is False, do Step 3 to 10

Step 3.0: For each Training pair, do Step 4 to 9

FEED FORWARD INPUT PATTERN

Step 4.0: Each input unit (x_i, i = 1, …, n) receives input signal x_i and broadcasts this signal to all units in the hidden layer.

Step 5.0: Calculate input signals to each hidden unit

$$z\_in_j = v_{oj} + \sum_{i=1}^{n} x_i v_{ij} \qquad 2(1)$$

Step 5.1: Activation function is applied to compute its output signal.

$$Z_j = f(z\_in_j) \qquad 2(2)$$

Then the signal is sent to all units in output layer.

Step 6.0: Each output unit (y_k, k= 1, ..., m) sums its weighted input signals.

$$y\_in_k = w_{ok} + \sum_{j=1}^{P} z_j w_{jk} \qquad 2(3)$$

Step 6.1: Apply activation function to compute its output signal.

$$y_k = f(y\_in_k) \qquad 2(4)$$

BACK PROPAGATION OF ERROR

Step 7.0: Each output unit (y_k, k= 1, ..., m) receives a target pattern. Corresponding to the input training pattern, computes its error information term.

$$\delta_k = (t_k - y_k)\, f^{1}(y\_in_k) \qquad 2(5)$$

Step 7.1: Calculate its weight correction term (used to update w_{jk} later)

$$\Delta w_{jk} = \alpha\, \delta_k\, z_j \qquad 2(6)$$

Step 7.2: Calculate its bias correction term (used to update w_{ok} later)

$$\Delta w_{ok} = \alpha\, \delta_k \qquad 2(7)$$

Send δ_k to units in the layer below

Step 8.0: Each hidden unit (z_j, j =1, ..., p) sum its delta inputs (from units in the output layer).

$$\delta\_in_j = \sum_{k=1}^{m} \delta_k w_{jk} \qquad 2(8)$$

Step 8.1: Multiply derivative of its activation function to calculate its error information term.

$$\delta_j = \delta\_in_j \, f^1(z\_in_j) \qquad 2(9)$$

Step 8.2: Calculate its weight correction term (used to update v_{ij}).

$$\Delta v_{ij} = \alpha \, \delta_j \, x_i \qquad 2(10)$$

Step 8.3: Calculate its bias correction term (used to update v_{oj}).

$$\Delta v_{oj} = \alpha \, \delta_j \qquad 2(11)$$

UPDATE WEIGHTS

Step 9.0: Each output unit (y_k, k= 1, …, m) updates its weights (j = 0, … , p):

$$w_{jk}(new) = w_{jk}(old) + \Delta w_{jk} \qquad 2(12)$$

Step 9.1: Each hidden unit (z_j, j =1, …, p) updates its weights (i = 0, …, n):

$$v_{ij}(new) = v_{ij}(old) + \Delta v_{ij} \qquad 2(13)$$

Step 10.0: Test for Stopping Condition

2.4.2 Back Propagation Activation Function

Basically there are two types of activation function implemented for back propagation network. An activation function for back propagation network must be continuous, differentiable, and monotonically non-decreasing and also the derivative is easy to compute (Fausett, 1994). One of the most usual activation function is the binary sigmoid function which is implemented if the range of the data is between 0 and 1 and is defined as:

$$f_1(x) = \frac{1}{1 + \text{exponential}\ (-x)} \qquad 2(14)$$

and it first derivative is :

$$f_1'(x) = f_1(x)\,[1 - f_1(x)] \qquad 2(15)$$

Another popular activation function for back propagation network is the bipolar sigmoid function, which is applied if the data range is between –1.0 to +1.0 and is defined as:

$$f_2(x) = \frac{2}{1 + \text{exponential}\ (-x)} - 1 \qquad 2(16)$$

with the activation function of

$$f_2'(x) = \frac{1}{2}\,[\,1 + f_2(x)\,][1 - f_2(x)] \qquad 2(17)$$

Chapter 3 Literature Review

According to Nicholas Findler 1991, the application of computer technology in medical started in the early 1960s. Computers have been widely used in medical domain to assist data gathering, collection, diagnosis, prognosis and selection of an appropriate treatment and medicine. One of the main purposes of computer application in medical field is to develop decision-support systems that can assist and enhance the clinician ability to diagnose, treat, and assess prognosis of pathologic conditions (Ohno-Machano, 1996). In late seventies and eighties, there were a lot of developments of medical expert systems in supporting the clinicians in their decision making especially in diagnosing certain disease. Probably the best-known medical information system is MYCIN, which was the first model being developed by Shortliffe (1976). MYCIN has 450 production rules in its knowledge and basically provides advice on the diagnosis and treatment of certain infectious disease, bacteremia and meningitis (Findler, 1991). Other efforts leading to specialized diagnostic and prognostic expert system such as CASNET (Weiss et al., 1978), GALEN (Thompson et al., 1983), INTERNIST (Miller, 1982) and CADUCEUS (Miller et al., 1984). Apart from being widely accepted as one the most important technology in medical decision support system, the above-mentioned expert systems have rule-bases, which are sometimes very hard and time consuming to build. In some cases, inconsistencies may arise when new rules are added to an existing database. There is also quite strong domain dependence in expert system; therefore knowledge bases can rarely be reused for new applications (Stensmo and Sejnowski, 1994). One of the bottlenecks in developing the expert system is the acquisition of knowledge because the scare of certain domains structured knowledge (Ohno-Machano, 1996). Nevertheless, expert systems are good in giving the explanations about the steps used to derive the best output.

In the late eighties and early nineties, new approaches to medical diagnosis have been discovered. Even though neural network has been introduced in the 1960s, it has been widely used in industry after the development of back propagation network (Rumelhart et al., 1986). Therefore, the emergence of back propagation algorithm brings new light and promise to medical application (Ohno-Machano, 1996). The ability of neural network to learn complex problem such as the causal relations among variable (symptoms) has increased the number of neural network applications in medical domain in the past few years compared to other methods.

3.1 Neural Network Application in Medical

Medical decision-making generally involves diagnosis and pattern recognition task. Artificial neural networks have been very successful particularly in such a task. Nowadays, neural networks have been vastly implemented in many areas in medical applications. For example, neural network has been trained to diagnose breast cancer (Ohno-Machado and Bialek, 1998). The results showed that the neural network models could be used as the alternative approaches in medical diagnosis. In another study, Delogu (1994) has conducted an automatic diagnosis of labeled monodispersed aerosols in obstructive lung disease. The study was conducted by using two back propagation methods, with momentum and the weight decay. The results of the study reveal that neural network classifications are very close to the human classification. In addition, neural network has also been widely used in the field of automatic sleep staging (Anderson and Sijercic, 1996; Huupponen et al., 1996; Groezinger et al., 1995; Sykacek et al., 1997). Apart form diagnosis, neural network has also been used in prognosis such as length of stay in the intensive care unit (Doig, 1993; Tu, 1993), survival after trauma (McGonical, 1993); admission to the psychiatry ward (Somoza, 1993). They found that neural network provided better prediction and prognosis in medical domain compared to statistical method such as logistic regression.

3.2 Neural Network Application in Cardiovascular Disorders and Heart Disease

One of the most widely used of neural network technology in medical applications is to diagnose and predict the cardiovascular disorders and heart disease. There are a lot of studies conducted in diagnosis, prognosis and prediction of heart disease, which is includes the coronary heart disease, myocardial infarction, chest pain and heart failure (Dorffner et al., 1994; Baxt, 1991; Simkus, 1994; Ortiz et al., 1995). Some of the examples are illustrated in the following section.

3.2.1 Diagnosis of Coronary Heart Disease

Dorffner et al., (1994), have conducted a study in evaluating the usefulness of neural network for early detection of coronary heart disease based on ECG and other measurement during exercise testing. Three different neural networks, two recurrents and one feed forward using background knowledge for processing were trained and compared to the performance of skilled cardiologist. The results of the study shown that neural network can compete with experts in classifying test as CHD or normal. The neural network has also outperformed classical statistical techniques published previously. Hence, the study indicates that the use of neural network improves the exercise of ECG as a non-invasive technique for detecting heart disease. Other researchers also conducted the same study (Akay, 1992; Akay et al., 1992; Dorffner and Koller, 1994) and found almost the same finding.

3.2.2 Diagnosis of Myocardial Infarction

A lot of studies have also been conducted to diagnose and predict myocardial infarction or heart attack (Baxt, 1991; Browner, 1992; Harrison et al., 1991a; Kennedy et al., 1990; McArthur et al., 1993; Downs et al., 1995; Harrison et al., 1991; Harrison et al., 1992). One of the most significant studies in diagnosing myocardial infarction is the one that was conducted by William Baxt (1991). In this study, neural network was trained on clinical pattern sets retrospectively derived from the cases of 351 patients hospitalized with a high probability of having myocardial infarction. It was prospectively tested on 331 consecutive patients presenting to an emergency department with anterior chest pain. The ability of the network to distinguish patients with from those without acute myocardial infarction was compared with that of physicians caring for the same patients. Surprisingly, neural network outperformed the physician in detecting the myocardial infarction.

The same work was conducted by Simkus (1994), but this study was made more challenging by having four choices instead of two; non-cardiac, angina, unstable angina or acute myocardial infarction. The study was based on 304 cases with 45 variables available in the emergency room. Simkus found that neural network performance was slightly greater than that for the clinician seeing the patients in the emergency room. He concluded that with the collection of further cases it is expected that the accuracy of the network will improve to the point that the method could become clinically useful. This information could then be used to support a decision whether a patient should be discharged or admitted and what level of treatment is required.

3.2.3 Mortality Prognosis in Heart Failure

Other studies concentrate on evaluating neural network technology in mortality prognosis in heart failure. Ortiz et al., (1995) sought to assess the usefulness and accuracy of artificial neural networks in the prognosis of 1-year mortality in patients with heart failure. Clinical and Doppler-derived echocardiographic data from 95 consecutive patients with diffuse impairment of myocardial contractility were studied. After 1 year, data regarding survival or death were obtained and produced the prognostic variable. Results of artificial neural network classification were compared with those from linear discriminant analysis, clinical judgment and conventional heuristically based programs. Ortiz concluded that the artificial neural network method has been proven to be reliable for implementing quantitative prognosis of mortality in patients with heart failure. Nevertheless, neural network are also used in detection and diagnosis of chest pain or angina (Kennedy et al., 1991; Marshall et al., 1991).

Chapter 4 Project Methodology

The purpose of this chapter is to give the detail description and discussion on the project methodology used to develop the application. Since the required data for the study is not available in a ready to use format, the application started with the development of Management Information System to provide a fundamental tools for data gathering and management. Once the data are collected, they must be prepared into certain format before being passed through the neural network. Before the development of neural network simulator, the neural network model and architecture must be defined. Then the network are trained and tested and the result are stored in the weight file, which was used by the prediction system. Finally, all works are integrated to form a single application model. Basically, there are seven major stages involved in building the application as listed below (see also Figure 4.1):

- The development of the Myocardial Infarction Management Information System (MIMIS)
- Data preparation
- Neural network modeling
- The development of Neural Network Simulator (NNS)
- Neural network training and testing
- The development of Prediction System (PS)
- The integration of all systems (MIMIS, NNS, PS)

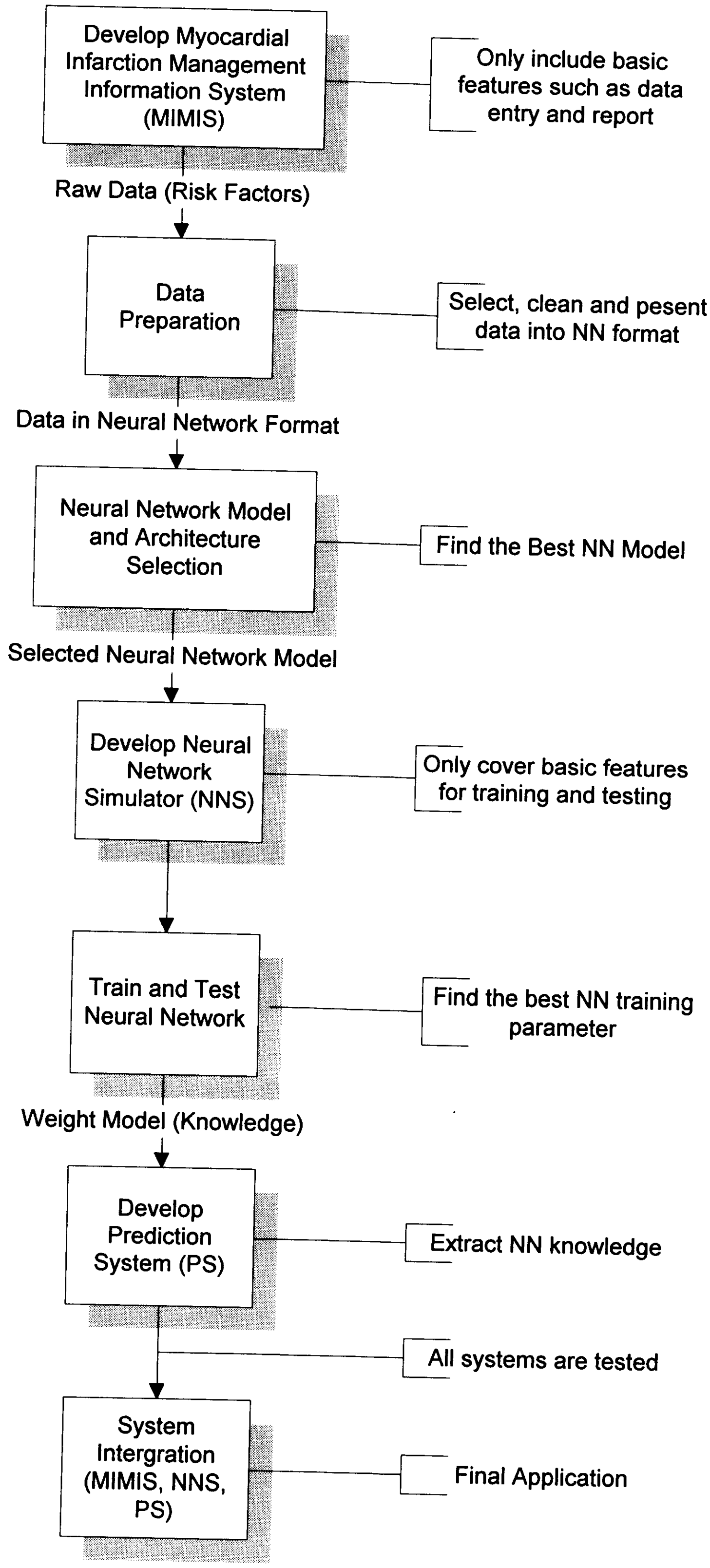


Figure 4.1: Major Steps in Developing the Application

4.1 The Development of the Myocardial Infarction Management Information System (MIMIS)

The development of the application began with the development of Myocardial Infarction Management Information System. The present condition in the Alor Setar General Hospital required the development of this system since it does not have any kind of computer information system. The development of this system is crucial because it serves as the initial step for data collection. Furthermore, this system is also used as the real system for data entry as well as the input for both Neural Network Simulator and Prediction System. Neural Network Simulator will extract the new data from this system for further retraining and retesting whilst the Prediction System uses new patient data for prediction purpose. The study is not meant for developing a comprehensive management information system but will include adequate facilities and tools for clinicians in managing patients' records. Basically, the system will have 3 main modules as describe below:

- Patient Registration
- Maintenance
- Reporting

4.1.1 Patient Registration Module

This is the main module for the system where it captures all patients' information such as personnel particular and risk factors. Apart from capturing data, the module also provides other facilities such as editing and searching patients' records. Users can use this module to search a particular patient by selecting the searching criteria such as the patient's id, or patient's name. Later, the module will be integrated with the Prediction System for automatic angina prediction.

4.1.2 Maintenance Module

Basically this module will have three sub modules, which are as follows:

- Code Maintenance
- Users Maintenance
- Neural Network Simulator

Code Maintenance

Since medical records involve a lot of technical terminologies and codes, it is important to provide tools for manipulating the codes.

User Maintenance

Mainly clinicians and system administrator will access the application. System administrator is responsible in maintaining users as well as the system configuration. System administrator has the full access for both the Myocardial Infarction Management Information System as well as the Neural Network Simulator.

Neural Network Simulator Maintenance

Apart form codes and users maintenance, the module will be integrated with the Neural Network Simulator. However, only the system administrator is authorized to access this module. The detail of the simulator is described in the later section.

4.1.3 Reporting

The module provides the users with some reporting tools such as the patient details report and medical report. The report is quite flexible where the users can have a lot of choices in selecting the report criteria.

4.2 Data Preparation

This is the most important part of the project development, however it is also the most difficult task to do. The patients' records are only available in patients' medical forms and are kept in the record department. Retrieving this kind of information involves a lot of procedures. Furthermore, the information must be translated into the questionnaire forms (see Appendix B) before the data entry. The doctors must do this kind of data conversion since the interpretation of medical record involves a lot of medical terminology. The study only managed to get 45 patients' records. The patients' information together with the details of the disease is entered into the Myocardial Infarction Management Information System, which has been developed before.

4.2.1 Data Selection

One of the most important parts in data preparation is to determine the best variables that contribute to the decision-making. The data selection step requires some detail knowledge of the problem domain and the underlying data (Bigus, 1996). Therefore, the selections of the variables are based on the advice from the doctors and also the review of the literatures. Even though there are quite a number of variables entered into the Myocardial Infarction Management Information System, only eight are identified as the important variables that contribute to the prediction of angina. Those variables are the major risk factors of angina. They are as follows:

- Age (Patient age)
- Family history of premature CAD (Yes or No)
- Smoking (Yes or No)
- Cholesterols (Cholesterol level)
- Hypertension (Yes or No)
- Diabetes mellitus (Yes or No)
- Hypercholesterolemia (Yes or No)

The description of each selected risk factors are described in the following section.

Age

Patient age and family history of premature CAD are considered as non-modifiable risk factors, which cannot be changed or modified. Generally if one gets older, the risk becomes higher. The risk is probably highest between sixty and seventy years (Mulcahy, 1979) because the arteries normally become harder and more rigid with age.

Family History of Premature CAD

Children of parents with heart disease are more susceptible to suffer from this disease. For example African Americans have more severe high blood pressure than whites and a higher risk of heart disease. This is partly due to higher rates of obesity and diabetes (Calnan, 1991). Most people with strong family history of heart diseases have one or more other risk factors

Smoking

Previous studies reveal that there is a strong relationship between cigarette smoking and heart disease (Doll and Peto, 1976). The greater the number of cigarettes currently smoked, the higher the risk of heart disease. Furthermore, based on research conducted by Doll and Peto (1976), the risk of fatal heart attack is two to three times higher for smoker compared to non-smoker. However, not all evidence supported the fact that smoking can increase the risk of heart disease. A study conducted by Cook et al., (1986), indicates that men who gave up smoking still have an increase risk of heart disease. Therefore, there is still considerable uncertainty about the mechanism that links cigarette smoking and heart disease (Calnan, 1991). Smoking cannot be claimed as an independent factor that contributes to the high risk of heart disease, however, combination with other factors such as blood cholesterols and blood pressure should be also considered.

Blood Cholesterol

Basically the elevated total blood cholesterol or elevated low-density lipoprotein (LDL) tends to be associated with high risk of heart disease. However elevated high-density lipoprotein (HDL) is a protective factor (Truswell, 1985). There was a continuous trend of increase risk of heart disease as the total cholesterol increase (Shaper et al., 1985). However, there is still uncertainty about whether blood cholesterol is causally associated with heart disease even some previous researches show the relationship. Generally, high cholesterol can increase the risk of heart disease, but this factor alone cannot spot the risk.

Hypertension

Hypertension is diagnosed by looking at the patient blood pressure. The people who developed hypertension usually have blood pressure toward the upper end of the range (Kannell, 1975). The study conducted by Reid et al., (1976) and Swales (1981), suggested that there is a strong relationship between elevated blood pressure (hypertension) and risk factor of heart disease. On the other hand, other studies show that there is no direct relationship between hypertension and heart disease. Combination of more than one factor such as hypertension and cholesterol seems to show the stronger relationship with heart disease (Gotto, 1988).

Diabetes Mellitus

Diabetes mellitus is the inability of the body to produce or respond properly to the hormone insulin. The body needs insulin to convert glucose (blood sugar) to energy. People with diabetes have higher ratios of total cholesterol to HDL cholesterol and also high incidence of hypertension, which lead to heart disease (Karnel and McGhee, 1979). Diabetes may cause lesions on the heart muscle that weaken it, as well as increase deposits of fat in the arteries. The study also show that combination of diabetes mellitus and other risk factors such as cholesterol and hypertension has a strong relationship to heart disease.

Hypercholesterolemia

Hypercholesterolemia is condition to indicate the high level of cholesterol in the blood. It is considered as the major risk factor for coronary heart disease that leads to heart attack. Generally, hypercholesterolemia can increase the risk of heart disease, but this factor alone cannot mark the risk (Shaper et al., 1985).

Kannel (1993) pointed out that there is no single factor found sufficient in the evolution of the heart disease. In other words, the risk associated with any one of these factors should not be calculated independent to the other factors (Percy, 1993).

4.2.2 Data Preprocessing

Once the potential data have been selected, the next step is to preprocess the data. According to Bigus, (1996), the preprocessing involves generating new data items from one or more data fields or something replacing several fields with a single field that contain the most significant value. Furthermore, the number of input fields does not closely related to the network performance. The first step that should be taken in data processing is to eliminate redundancy, since it can reduce the network performance (Bigus, 1996).

Initially, there are seven related risk factors associated for each input. Hypercholesterolemia and cholesterol level represent the same thing. Hypercholesterolemia is a condition to indicate the high level of cholesterol in patients. Cholesterol level on the other hand measures the level or cholesterol. Therefore, cholesterol level by itself can indicate the hypercholesterolemia. Due to these facts, the number of risk factors for each input is reduced from 7 to 6 only.

Most neural network models only accept numeric data in the range of 0.0 to 1.0 or –1.0 to +1.0. Therefore, certain data must be scaled down within that range.

In this study, there are two data fields that have continuous numeric values, namely the patient's age and the cholesterol level. In these cases, both data fields must be normalized. According to Bigus (1996), there are three types of normalization method commonly used as describe below:

- Method 1: Sum the squares of each element, then take the square root of the sum, and finally divide each element by the norm.
- Method 2: Simply sum up all the elements in the vector and then divide each number by the sum. In this method, the normalized element sum to 1.0 and each takes on a value representing the percentage of contribution they make.
- Method 3: Divide each vector element by the maximum value in the array.

For simplicity, method 3 has been adopted in this study. Others data fields such as the smoking, diabetes mellitus, family history of premature CAD and hypertension also need to be presented in neural network format.

4.2.3 Representation

Data representation is important since if wrong decisions are made regarding the representations, the neural network will not learn anything (Bigus, 1996). This study only involves two kinds of data types. The first one is the continuous numeric, which has been normalized above. The second data type is the categorical, which only has two values, YES and NO. In this case, the value of YES is represented by '1' and NO is changed to '0'. The final data type and data presentation is listed in Table 4.1.

Table 4.1: Data Type and Presentation

Field Name	Data Type	Values	Representation and Normalization Method
Age	Continuous Numeric	0 to Maximum value	Scale 0 to 1 New Value = Old Value / Max Value
Family History of Premature CAD	Categorical	YES, NO	YES = 1 NO = 0
Smoking	Categorical	YES, NO	YES = 1 NO = 0
Cholesterol level	Continuous Numeric	0 to Maximum Value	Scale 0 to 1 New Value = Old Value / Max Value
Hypertension	Categorical	YES, NO	YES = 1 NO = 0
Diabetes Mellitus	Categorical	YES, NO	YES = 1 NO = 0

4.3 Selection of Neural Network Model and Architecture

Once the data is gathered, preprocessed, the data is represented in the form that is suitable for neural network format. The next step is to select the best neural network model and architecture to use in the study. Before selecting the model and architecture, the learning paradigm and topology of the network must be identified.

4.3.1 Selection of Neural Network Learning Paradigm

Normally the selection of learning paradigm is based on the nature of the problem. Basically there are three main learning paradigms, they are supervised, unsupervised and reinforcement learning (Bigus, 1996). Supervised learning is best for prediction and classification, while unsupervised is suitable for clustering and finally reinforcement is useful in optimization over time. Since the purpose of the study is to predict the presence of angina in patients, the supervised learning paradigm is selected.

4.3.2 Selection of Neural Network Topology

After selecting the training paradigm, the next thing to do is to select the neural network topologies since it can give a big impact on the processing capabilities of the neural networks. Different neural network topology resulted in different network performance. Normally there are three major connection topologies that define how the data pass through the input, hidden and output nodes (Bigus, 1996). They are the feed forward, limited recurrent and fully recurrent networks. The feed forward is preferred if the data flows through the network in one direction and the answer based solely on the current set of input or problems. The limited recurrent on the other hand required the network to store a record of prior inputs and factor them in with the current data to generate the output. Finally, the fully recurrent networks provides a two-way connections between all processors and data flows to all adjacent units back and forth until the activation of the units stabilizes (Bigus, 1996). In this study, all problems or inputs pass through the network in one direction and the answer can be generated based on current inputs only. Therefore, the feed forward network topology is selected.

4.3.3 Selection of Neural Network Learning Algorithm

Once the training paradigm and the network topology have been chosen, the next step is to select the learning algorithm. There are various types of learning algorithms available, however only four of them are suitable for the supervised training paradigm with feed forward network topology and classification purposes. These learning algorithms are the back propagation, the radial basis function networks, the probabilistic neural network as well as the learning vector quantization (Bigus, 1996). Tentatively, the back propagation is chosen as the learning algorithm for the feed forward network, however the final decision depends on other factors such the presentation of the data.

4.3.4 Selection of Neural Network Model

In order to select the best network model, further steps are carried out that mainly involved the presentation of the data and learning requirement (Bigus, 1996). In general, the best model and architecture selection is based on the following two issues:

- Data type

  Once the function to perform has been identified, the network topology as well as the learning paradigm, the next step is to look at the data types. Data types for this study are all in binary format (values ranges between 0.0 to 1.0).

- Training requirement

  Finally the learning or training requirement is identified. Basically training speed becomes the overriding factor in determining which neural network to use for an online learning (Bigus, 1996). When a fast and online learning situation is required, probably the Probabilistic Neural Network or Radial Basis Function will be our best choice. However, the training requirement for this project needs neither fast nor an online training.

After considering all issues mentioned above, back propagation has been chosen as the best network model. The selection also agrees with the previous studies where most of the studies employed back propagation network in their application of neural network in clinical diagnosis (Fraser et al., 1994; Tsai, Y.S. et al., 1990; Delogu, 1994; Ohno-Machado and Bialek, 1998). Apart from that, the back propagation network is chosen since it is the most widely accepted and a lot of literatures related to this model. In addition, programming codes for back propagation are easily available either from the book or from the Internet

4.3.5 Definition of Input and Output Nodes

Once the neural network model is selected, the next step is to define the input and output for the network. Normally, the definition of input and output nodes is determined by the data presentation. Inputs variables for neural network are the units that receive signals from outside the neural network and they normally send the input signal to all neurons to which they are connected without modification (Fausett, 1996). Basically, the inputs for neural network are those variables that determine the output or target value. In this study, there are eight variables that can influence the results or outputs. Those variables are the risk factors that determine the presence of angina (output).

Neural Network Inputs

The variables (risk factors) associated with each input are listed as follows:

- Age (Patient age)
- Family history of premature CAD
- Smoking
- Hypertension
- Diabetes mellitus
- Cholesterols

Neural Network Output

As for the output, there are only two possibilities, namely the presence of angina or the absence of it.

- Presence of Angina

4.4 The Development of Neural Network Simulator

Once the model and architecture of the neural network are chosen, the next step is to train and test neural network with the selected data. The training and testing require the neural network simulator. Even though there are many kinds of commercial neural network software available in the market, a neural network simulator based on the model and architecture was developed. The main purpose of the development of this simulator is to extract the trained neural network model weights for predicting the new data set. However, some commercial software provides the model weights of the trained neural network, but it involves a lot of interpretations. On the other hand, the model weight written by the Neural Network Simulator in this is presented in easy-to-read and ready to use without requiring a lot of interpretation. The neural network simulator uses the back propagation-learning algorithm, which is not differing from other commercial software.

4.4.1 Selection of Activation Function

The activation function of binary sigmoid is implemented in this neural network simulator since the input and output data are all in binary format, which is in the range of 0 to 1.

4.5 Neural Network Training and Testing

Once the data preparation is complete and the neural network model and architecture is selected, the next important step is to train the neural network. Then the trained network has to go through testing phase to see the network performance. The main purpose of neural network training is to teach the network to make good generalization about new data set which the network never seen before. Therefore, the network should not memorize the network patterns, but must learn the relationship among the variables involves in the network. The performance of the network is based on the testing results and the training result. The network with the highest testing results and with the lowest training results is considered as the best network model.

Normally the neural network training and testing involves few stages as described in the Figure 4.2. It stated with the selection of the best-hidden unit, followed by selection of learning and momentum rate, the activation function and finally the best stopping criteria is defined.

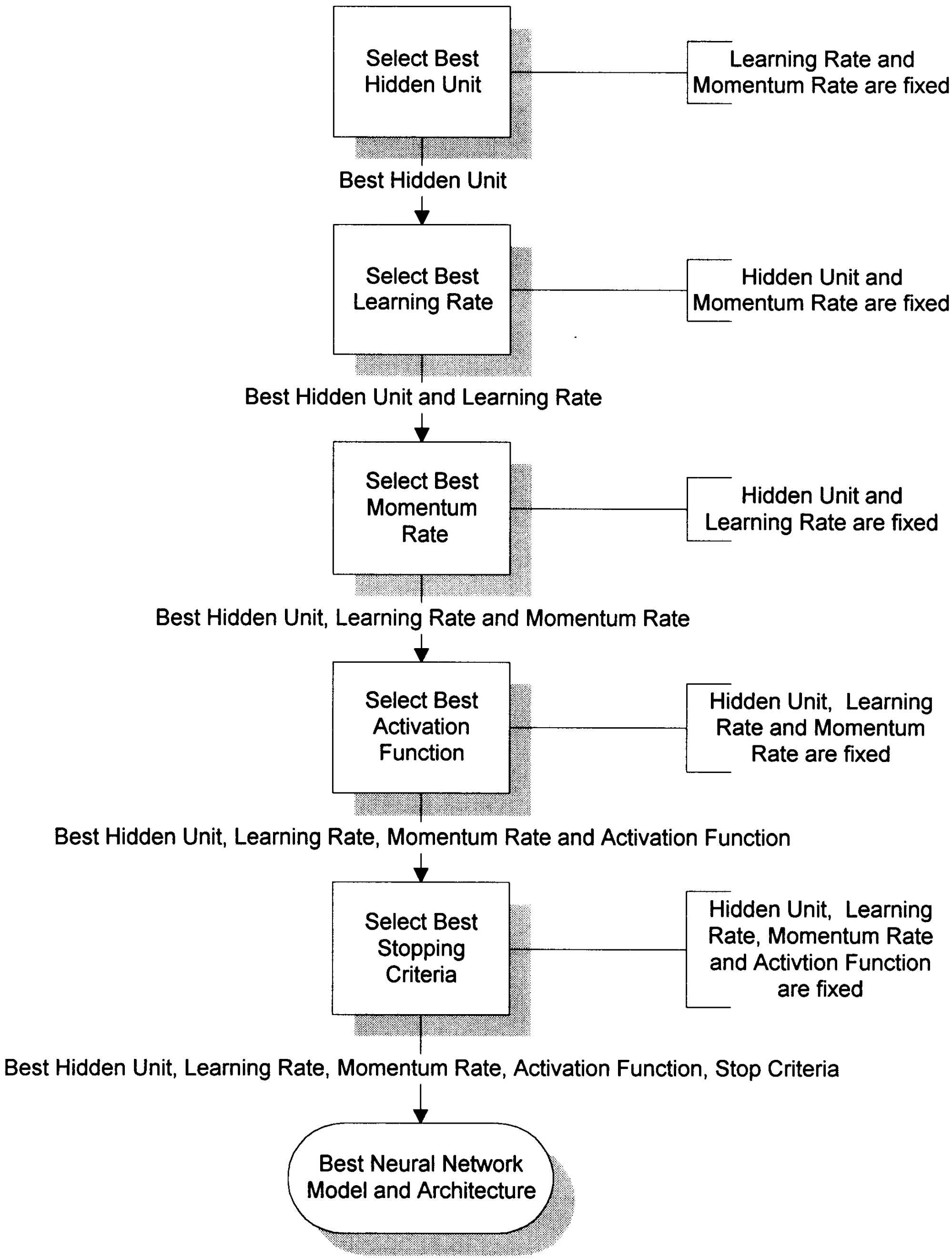


Figure 4.2: Major Steps in Neural Network Training and Testing

4.5.1 Selection of Hidden Unit

Based on the previous studies, there was no specific rule in determining the number of hidden nodes at the hidden layer. Normally, the best number of hidden nodes can be obtained by trial and error testing (Tsoukalas and Uhrig, 1997). The "rules of thumb" for choosing the hidden unit is that; sum the number of inputs plus the number of outputs divided by two. However many people argue that there is no way to determine a good network hidden unit just from the number of inputs and outputs. It might depend critically on the number of training cases, the amount of noise, and the complexity of the function or classification. There are problems with one input and one output that require a lot of hidden units, and problems with a many inputs and a many outputs that require only one hidden unit. Basically, it all depends on the problem and also the training cases.

Therefore, in this study, a set of hidden units is tested, which range from hidden unit 2 up to hidden unit 11. Then the best two hidden units are selected and once again are tested with different seed number to come out with the best-hidden units.

4.5.2 Selection of Learning Rate

Most of neural network models have learning rate parameters associated with them. The learning rate is considered as the key that controls the learning paradigm. Basically the learning rate parameter controls the magnitude of the changes in adjusting the connection weights. A large learning rate causes the big changes in the weights after each epoch and may lead to giant oscillations in their values. Apart from that, the purpose of training is not to get the actual answer for each training pattern; besides, the network is trained to generalize the unfamiliar patterns. Generally a lower learning rate is chosen since it will learn step by step and finally be able to make good generalization.

In this study, in order to cover all the possibilities, learning rate in the range of 0.1 to 1.0 is being tested. Then the best two learning rates are selected and once again are tested with different seed number. Finally the best learning rate is chosen.

4.5.3 Selection of Momentum Rate

Momentum rate is always go hand in hand with the learning rate and it will filter out high-frequency changes in the weights. Therefore, the network will not stated oscillating around a set of values (Bigus, 1996). The weights are less likely to be driven up and down in alternate directions. The errors from the previous training patterns are averaged together overtime and added to the current error. So if the error for a particular pattern is too big and in the opposite direction, it will not force the network to change abruptly in that direction because the error will be averaged in.

In this study, in order to cover all the possibilities, momentum rate in the range of 0.1 to 1.0 is being tested. Then the best two momentum rates are selected and once again are tested with different seed number. Finally the best momentum rate is chosen.

4.5.4 Selection of Activation Function

Basically there are two kinds of activation function normally used in back propagation network, which are the bipolar sigmoid and the binary sigmoid. The selection of activation function is based on data representation. If the data is binary, binary sigmoid should be selected whereas if the range is in between –1 to +1, bipolar sigmoid must be implemented. The training and testing data for this study is binary, therefore, the binary sigmoid is selected.

4.5.5 Selection of Stopping Criteria

One of the crucial steps in training a neural network is to understand when to stop (Bigus, 1996). There are two types of stopping criteria implemented in this Neural Network Simulator. The first one is to compare the error tolerance with the total Mean Square Error (MSE). Neural network will stop training when it reaches the situation where the total MSE is equal to the error tolerance specified. In the study, the error tolerance is set to 0.0001. The second condition is to stop training when the network reaches the maximum epoch or iteration. One epoch is defined as a single pass through a complete data set while training or testing the neural network. Intuitively, one might think that if 100 epochs is good, then 1000 epochs will be much better. In the other words, one generally think that the more the training the better the performance of the network. However, this situation is not true in neural network (Bigus, 1996). If the same input patterns were presented to the neural networks over and over again, the neural network might perform extremely well on the training phase, but it cannot generalize with the new input pattern during the testing phase. This condition is known as over training. In this case the network tends to memorize the patterns, rather than learn the relationship among the variables. It is important to realize that the study is not trying to get the best prediction result for training, but what is more important is the ability of the network in predicting the testing data.

In this study, a wide range of epoch number is tested in order to cover all the possibilities. An epoch of 100 is chosen as a starting point, then followed by 500, 1000, 1500, 2000, 2500, 3000 and 3500 epochs.

4.5.6 Use of UCI Database for Comparison

Cleveland data set from the UCI Machine Learning database has been chosen for comparison. Basically, the inputs and the outputs for the network are described as follows:

Neural Network Input

The input for UCI Machine Learning database consists of:

- Patient age (continuous numeric)
- Trestbps: Resting blood pressure in mm Hg on admission (continuous numeric)
    - To indicate Hypertension
- Chol: Serum Cholesterol in mg/dl (continuous numeric)

- Cigs (cigarettes per day) (continuous numeric)
  - If '0' means non smoking, else is a smoking
- Fbs: (fasting blood sugar > 120 mg/dl) (yes, no)
  - To Indicate Diabetes
- FamHis: Family History of CAD (yes, no)

Neural Network Output

As for the output, there are only two possibilities, namely the exercise induces angina or the exercise will not induce angina.

- Exang: Exercise induces angina (yes, no)

Basically, there are 185 records selected from Cleveland dataset. 80% is used for training and another 20% for testing. The dataset also gone through the data preparation and presentation process before feed into the network. It also follows the same standard for training and testing as for previous dataset.

4.6 The development of Prediction System

Once the performance of the neural network is satisfied, the next step is to deploy and use it to score new patient's data. The development of Prediction System involves 4 major steps, which are as follows:

Step 1: Reading input parameters from Myocardial Infarction Management Information System.

Step 2: Automatic data preprocessing and representation.

Step 3: Extract the model weights from trained neural network and feed forward the inputs through the network.

Step 4: Calculate the output and post processing (scale) it for prediction purposes.

Step1: The inputs for the prediction system are the information captured from the patient registration module. Only certain variables are selected and these variables are exactly the same as the input variables for training and testing the neural network. However, in this prediction module, the target value, which is the presence of angina in patients, is not provided. Besides, the system will predict the target based on the inputs values and the neural network knowledge.

Step 2: Since the inputs are the not normalized yet, they need to be preprocessed and represented before they can pass through the neural network. The preprocessing technique implemented in this module is exactly the same as the one that implemented in the neural network training. All values must be in the range of 0.0 to 1.0.

Step 3: Once new patient's data are selected and preprocessed, the next step is to feed these values into the neural network. The prediction process only required the feed forward phase since no further training and weight adjustment needed. The same feed forward application programming that used in neural network training is implemented, however, there are two sections that has to be changed. First, instead of assigning random values between –0.5 and 0.5 to the initial weights, the prediction module reads the trained neural network weights from the weight file. The weights are fixed and will not change anymore. Second, the inputs must be automatically preprocessed.

Step 4: Post processing is a transformation of the output value from neural network format to the value that is meaningful to the clinicians. The calculation of neural network produces the output value in the range of 0.0 to 1.0. Since the prediction only has two kinds of possibilities, the neural network produces two kinds of bin as follows:

- Bin 1: Value ranges form 0.0 to 0.494
- Bin 2: Value ranges form 0.495 to 1.0

Then transformation process translates the Bin1 to 'has an angina' whereas Bin2 to 'has no angina'.

4.7 The System Integration (MIMIS, NNS and PS)

Once the individual system completed and tested, it will be integrated into one single system. For the purpose of the study, the Myocardial Infarction Management Information System will serve as the main system for the integration. The Neural Network Simulator is integrated with the Maintenance Module whilst Prediction System is integrated with the Registration Module. The data flow of the final system is described in the following diagram.

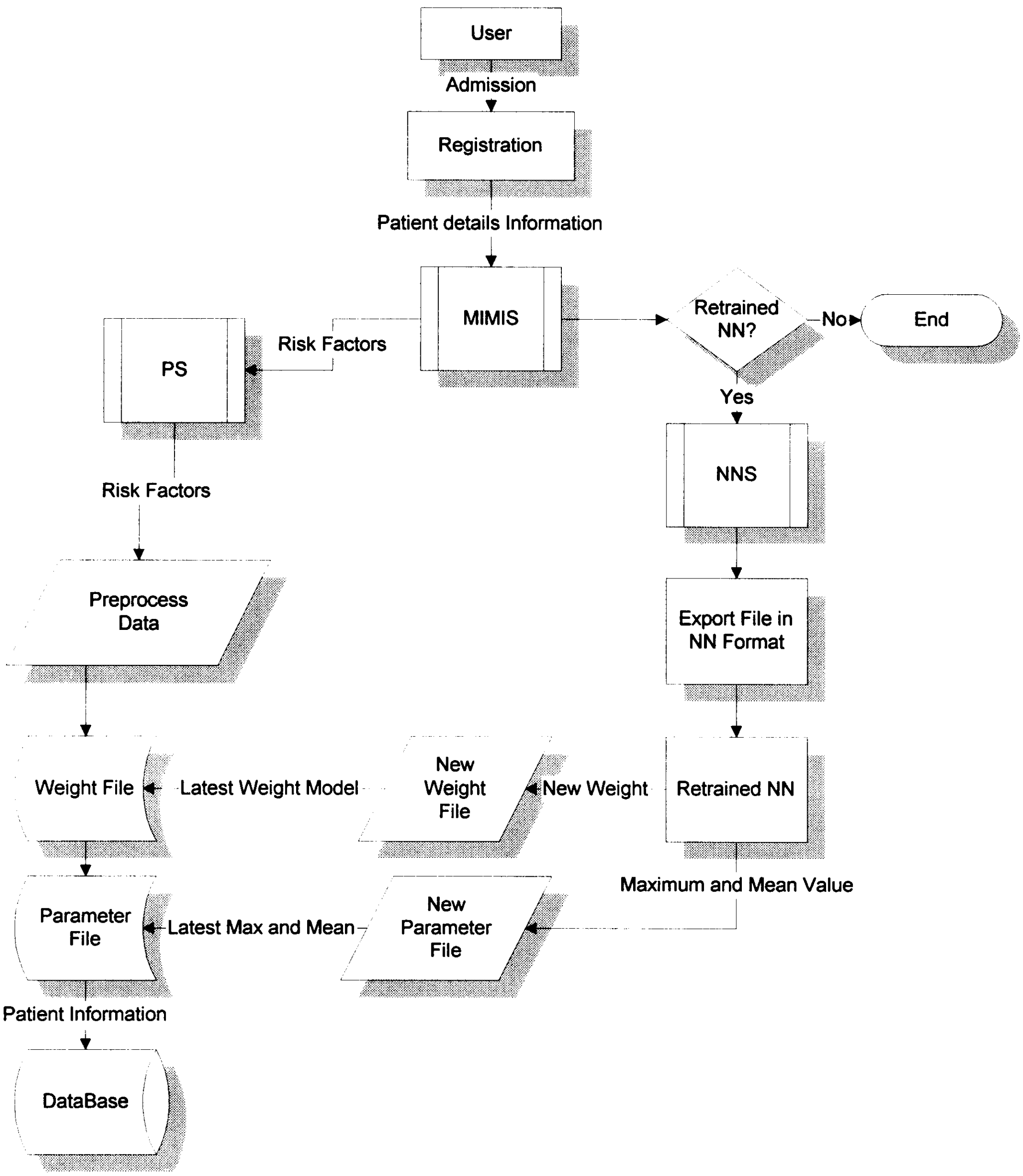


Figure 4.3: Final System's Data Flow

Chapter 5 Findings and Results

In this chapter, the results for each training and testing is described in details. Finally, the best neural network model is presented.

Table 5.1: Selection of the Hidden Units

HU	TRAINING	TESTING
2	50.00	44.44
3	50.00	44.44
4	83.33	77.78
5	83.33	88.89
6	86.67	77.78
7	86.67	77.78
8	83.33	88.89
9	86.67	77.78
10	86.67	77.78
11	83.33	88.89

Hidden Unit 5 and 8 (both with training result of 83.33% and testing accuracy of 88.89%) have been chosen since they produce the highest testing accuracy and with the lowest possible training result. Hidden Units 11 also shows the same performance of training and testing, but has not been chosen because it require more training and testing resources compared to Hidden Unit 5 and 8 (Bigus, 1996).

Table 5.2: The Best Hidden Unit				
SEED	HU-5-TRAIN	HU-5-TEST	HU-8-TRAIN	HU-8-TEST
1	83.33	88.89	83.33	88.89
2	83.33	77.78	83.33	88.89
3	83.33	77.78	83.33	77.78
4	86.67	77.78	86.67	77.78
5	86.67	77.78	83.33	88.89
6	83.33	77.78	86.67	77.78
7	86.67	77.78	86.67	77.78
8	86.67	77.78	86.67	77.78
9	83.33	77.78	83.33	88.89
10	86.67	77.78	86.67	77.78
AVE	85.00	78.89	85.00	82.22

Hidden Unit 8 (average training of 85.00% and testing of 82.22%) has been chosen since it produces the same training average but at the same time has a higher testing result compared to Hidden Unit 7 (average training of 85.00% and testing of 78.89%).

Table 5.3: The Selection of Learning Rates		
LEARNING-RATE	HU-8-TRAIN	HU-8-TEST
0.1	83.33	88.89
0.2	93.33	77.78
0.3	90.00	77.78
0.4	90.00	77.78
0.5	86.67	77.78
0.6	90.00	77.78
0.7	90.00	77.78
0.8	90.00	77.78
0.9	86.67	77.78
1.0	86.67	77.78

Learning Rate 0.1 (training of 83.33% and testing of 88.89%) and Learning Rate 0.5 (training of 86.67% and testing of 77.78%) have been chosen since they produce good testing result with the lowest training result. Even though

Learning Rate 0.9 and 1.0 are also produce the same training and testing accuracy, they are not chosen since higher learning rate can lead to giant oscillations (Bigus, 1996).

Table 5.4: The Best Learning Rate

SEED	LR-0.1-TRAIN	LR-0.1-TEST	LR-0.5-TRAIN	LR-0.5-TEST
1	83.33	88.89	86.67	77.78
2	83.33	88.89	90.00	77.78
3	83.33	77.78	90.00	66.67
4	86.67	77.78	86.67	77.78
5	83.33	88.89	90.00	77.78
6	86.67	77.78	86.67	66.67
7	86.67	77.78	90.00	66.67
8	86.67	77.78	90.00	77.78
9	83.33	88.89	90.00	66.67
10	86.67	77.78	90.00	77.78
AVE	85.00	82.22	89.00	73.34

When compared the overall performance of Learning Rate 0.1 (average training of 85.00% and testing of 82.22) and Learning Rate 0.5 (average training of 89.00% and testing of 73.34), the first produces higher accuracy on testing while at the same time produces lower training result. Therefore, the Learning Rate of 0.1 is chosen.

Table 5.5: The Selection of Momentum Rates		
MOMENTUM-RATE	TRAIN	TEST
0.1	83.33	88.89
0.2	83.33	88.89
0.3	83.33	88.89
0.4	83.33	88.89
0.5	83.33	88.89
0.6	83.33	77.78
0.7	83.33	77.78
0.8	86.67	77.78
0.9	86.67	77.78
1.0	86.67	77.78

Neural Network with the Momentum Rate of 0.1 and 0.5 (both with the training accuracy of 83.33% and testing result of 88.89%) have been selected since they produce best testing result with the lowest possible training result.

Table 5.6: The Best Momentum Rate				
SEED	MR-0.1-TRAIN	MR-0.1-TEST	MR-0.5-TRAIN	MR-0.5-TEST
1	83.33	88.89	83.33	88.89
2	83.33	77.78	83.33	88.89
3	83.33	88.89	83.33	77.78
4	83.33	77.78	86.67	77.78
5	83.33	88.89	83.33	88.89
6	83.33	88.89	86.67	77.78
7	83.33	88.89	86.67	77.78
8	83.33	88.89	86.67	77.78
9	83.33	88.89	86.67	77.78
1.0	83.33	77.78	86.67	77.78
AVE	83.33	85.56	85.33	81.11

The Momentum Rate of 0.1 (average training of 83.33% and testing of 85.56%) is selected since it produces higher testing result with lower training result compared to Momentum Rate of 1.0 (average training of 85.33% and testing of 81.11%).

Table 5.7: Selection of Stopping Criteria (EPOCH)		
EPOCH	TRAIN	TEST
100	80.00	77.78
500	80.00	77.78
1000	83.33	88.89
1500	86.67	88.89
2000	86.67	77.78
2500	86.67	77.78
3000	93.33	77.78
3500	93.33	77.78
4000	93.33	77.78
4500	93.33	77.78

The neural network training must be stopped whenever it started to memorize the input patterns. From the above table, the network performs best at 1000 epoch or iteration. Figure 5.1 illustrates the over fitting occurs at 2500 epochs. Even though the training result became higher, the testing result on the other hand started to decline.

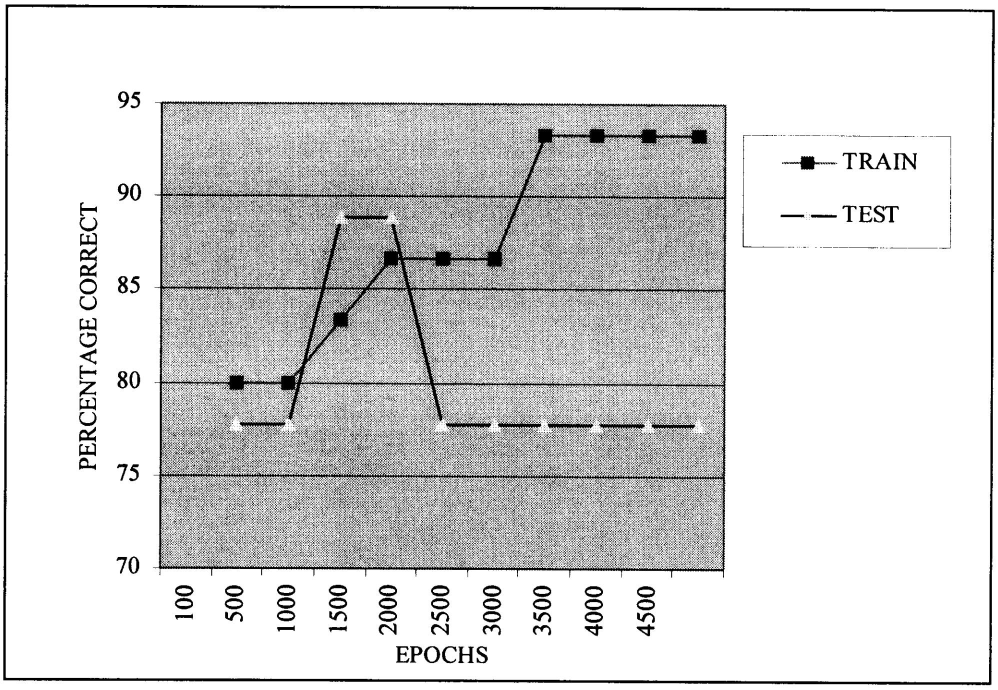


Figure 5.1: Stopping Criteria

Finally, the best network model obtained consists of 6 input units, one hidden layer with 8 hidden units and one output unit. The learning rate of 0.1 and the momentum rate of 0.1 have been selected. The best stopping criteria is at 1000 epochs. The network performed best with binary sigmoid activation function. The final result is presented in Table 5.8. The network model has been able to perform quite well with the prediction accuracy of 88.89% when presented with the testing data set.

As with the previous dataset, the Cleveland dataset also shows good prediction accuracy with the selected neural network model. It produces a training result of 80.64% and is able to predict at the accuracy of 90.00%

Table 5.8: The Final Model	
Model	Multi Layer Perceptron with Back Propagation Algorithm
Input Unit	6
Hidden Layer	1
Hidden Unit	8
Output Unit	1
Learning Rate	0.1
Momentum Rate	0.1
Activation Function	Binary Sigmoid
Stopping Criteria	1000 Epochs

The final neural network model implemented in described in the following figure.

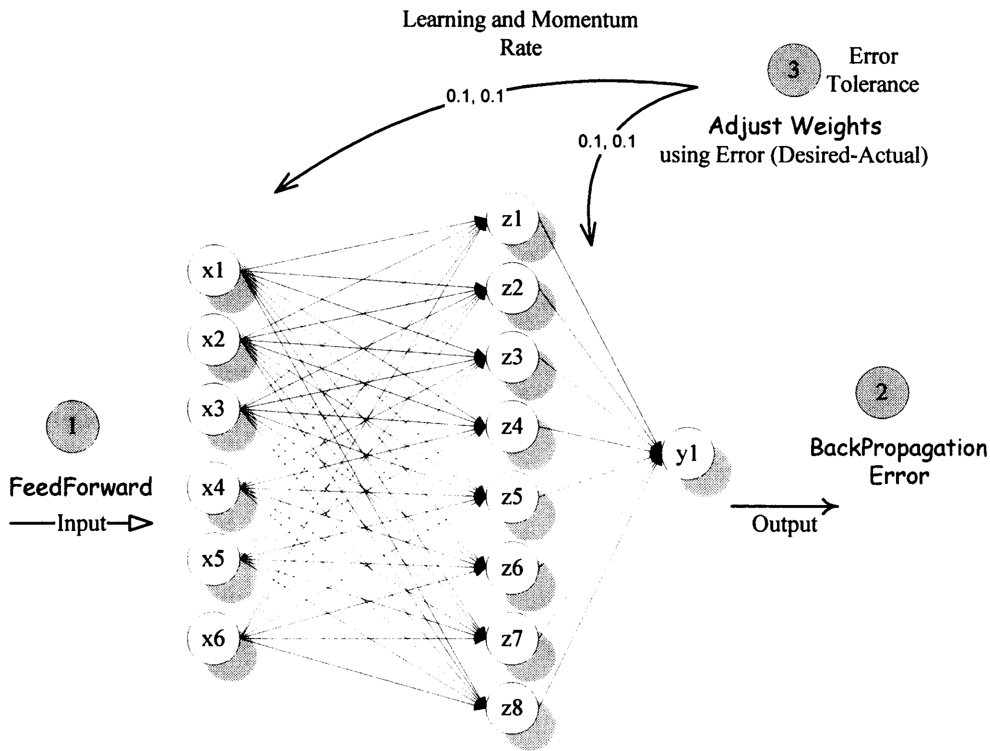


Figure 5.2: Neural Network Model Implemented

The inputs for neural network are as follows:

- x1 = Age (Patient age)
- x2 = Family history of premature CAD

- x3 = Smoking
- x4 = Hypertension
- x5 = Diabetes mellitus
- x6 = Cholesterols

The neural network only has one output as follows:

- y1 = Presence of Angina

Chapter 6 Conclusion and Recommendation

Basically, the main objective of the study, which is to evaluate the application of neural network technology in diagnosing the heart disease, particularly the presence of an angina in patients has been successful. The performance of back propagation neural network with the prediction accuracy of 88.89% is satisfactory. With the high accuracy of prediction, neural network model can be applied in medical application particularly in predicting the angina in patients. However, the current prediction system is not reliable enough because of the lack of patient's data. There are more cases that need to be considered, trained and tested by the application. With the collection of further patient's data, it is expected that the accuracy of the network will improve and at the same time increase the prediction reliability up to the point where the method could become clinically useful. This information could then be used to support a decision whether or not a patient should go through the EST. Apart from proving the ability of neural network technology in medical diagnosis, this study also shows how the neural network can be integrated into a management information system as a prediction tools. The application developed can be used as the starting point in building a medical decision support system, particularly in diagnosing the heart disease.

The selections of the risk factors are based on the literature reviews and also being supervised by the doctor. However, the risk factors for the study are limited to the variables that are stated in the questionnaire form (see Appendix B). People could argue that, there might be other risk factors, which could lead to heart disease or angina. Other contributing factors such as the obesity (Higgins, 1993), gender (Jenner, 1993), elevated hematocrit (Gagnon, 1994) and serum fibrinogen (Kannel, 1992) should also been considered.

The Neural Network Simulator only implemented the back propagation algorithm with the binary sigmoid activation function. In future, further study could be carried out to test different neural network models such as the Radial Basis Function as for comparison.

In conclusion, this study has succeeded in initiating a pilot project that utilized neural network in predicting the presence of an angina in patients.

Bibliography

Aho, A.V., Hopcroft, J.E. & Ullman, J.D. (1974), *Design and Analysis of Algorithm*, Addison Wesley, Reading, MA

Akay, M. (1992), *Noninvasive Diagnosis of Coronary Artery Disease Using a Neural Network Algorithm*, Biol. Cybern., 67(4):361-7

Akay, M., Akay, Y.M. & Welkowitz, W. (1992), *Neural Networks for the Diagnosis of Coronary Artery Disease*, Proceed. Int. Joint Conf. Neural Networks, Baltimore, MD, New York: IEEE Press

Amari, S.I. & Arbib, M.A. (1982), *Competition and Cooperation in Neural Nets*, Springer-Verlag, New York

American Heart Association (2000), [www document]. URL. http://www.americanheart.org/Heart_and_Stroke_A_Z_Guide/cvds.html

Ballard, D.H. (1986), *Cortical Connections and Parallel Processing*, Behav. Brain Sc., 9(1), 67

Baxt, W.G. (1991), *Use of an Artificial Neural Network for The Diagnosis of Myocardial Infarction*, Ann. Intern. Med., 115: 843-848

Bigus, J.P. (1996), *Data Mining with Neural networks: Solving Business Problems - from application development to decision support*, McGraw Hill, New York

Browner, W.S. (1992), *Myocardial Infarction Prediction by Artificial Neural Networks*, Ann. Intern. Med., 116(8):701-2

Calnan, M. (1991), *Preventing Coronary Heart Disease: Prospects, Policies and Politics*, Routledge, London

Cook, D.G., Pocock, S.J., Sharper, A.G. & Kussick, S.J. (1986), *Giving Up Smoking and the Risk of Heart Attacks*, Lancet, 11, 1376-9

DARPA (1988), *Neural Network Study*, AFCEA International Press, New York

Delogu, P. (1994), *Neural Network for Automatic Diagnosis of Labeled Monodispersed Aerosols in Obstructive Lung Disease*, Dipartimento di Fisica dell' Universiti and sezione I.N.F.N., Pisa, Italy

Doig, G.S., Inman, K.J., Sibbald, W.J., Martin, C.M. & Robertson, J.M. (1993), *Modeling mortality in the intensive care unit: comparing the performance of a back-propagation, associative-learning neural network with multivariate logistic regression*, in InSafran C(ed), Proceedings of the Seventeenth Annual

Symposium on Computer Applications in Medical Care, McGaw-Hill, New York

Doll, R. & Peto, R. (1976), *Mortality in Relation to Smoking*, British Medical Journal, 2, 1523-36

Dorffner, G. & Koller, G. (1994), *Toward Improving Exercise ECG for Detecting Ischemic Heart Disease with Recurrent and Feedforward Neural Nets*, Austrian Research Institute for Artificial Intelligence, Vienna, Austria

Downs, J., Harrison, R.F. & Kennedy, R.L., (1995), *A prototype neural network decision-support tool for the diagnosis of acute myocardial infarction*, Proceedings of the European Society for Artificial Intelligence in Medicine Fifth European Conference on Artificial Intelligence in Medicine, AIME-95, Pavia, Italy, 355-366

Fausett, L. (1994), *Fundamentals of Neural Networks: Architectures, Algorithms and Applications*, New Jersey: Prentice Hall

Findler, N.V. (1991), *An Artificial Intelligence Technique for Information and Fact Retrieval: An Application in Medical Knowledge Processing*, MIT Press, England

Gotto, A.M. (1988), *Co-existent Hypercholesterolemia and Hypertension in CHD Risk: Clinical Aspects and Considerations*, Lipid Review, 2, 3, 17-22

Harrison, R.F., Marshall, S.J. & Kennedy, R.L. (1991a), *A connectionist aid to the early diagnosis of myocardial infarction*, Proceed. Third European Conf. Artificial Intelligence in Medicine, Maastricht, The Netherlands

Harrison, R.F., Marshall, S.J. & Kennedy, R.L. (1991), *The early diagnosis of heart attacks: a neurocomputational approach.* Proceedings of the Institute of Electrical and Electronics Engineers and the International Neural Network Society International Joint Conference on Neural Networks, Seattle, 1-5

Kannel, W.B. (1993), *Hypertension as a risk factor for cardiac events, epidemiologic results of long-term studies*, Journal of Cardiovascular Pharmacology, 21 Suppl 2:S27–37.

Karnel, W. & McGhee, D. (1979), *Diabetes and Cardiovascular Disease: The Framingham Study*, JAMA , 241:2035

Karnell, W.B. (1975), *Risk of Blood Pressure in Cardiovascular Disease: The Framingham Study*, Angiol, 26, 1-14

Kennedy, R.L., Harrison, R.F., Kirklis, K., Moriarty, K.T., Gangi, M.T., Shaukat, H.N., & Young, M.J. (1990), *Early diagnosis of acute myocardial infarction: A novel approach to decision making using neural networks*, Clinical Science, 78, 88-88

Kennedy, R.L., Harrison, R.F., Marshall, S.J. & Hardisty, C.A. (1991), *Analysis of clinical and electrocardiographic data from patients with acute chest pain using a neurocomputer*, Quarterly Journal of Medicine, 80, 788-789

Kohonen, T. (1977), *Associative Memory: System Theoretic Aproach*, Springer-Verlag, Berlin

Marshall, S.J., Harrison, R.F. & Kennedy, R.L. (1991), *Neural classification of chest pain symptoms: A comparative study*. Proceedings of the Institution of Electrical Engineers Second International Conference on Artificial Neural Networks, Bournemouth. 200-204

McArthur, D., Hamer, W.G., McAllum, R., Harrison, R.F., Kennedy, R.L. & Steedman, D.J. (1993), *Early diagnosis of acute myocardial infarction (AMI): comparison and combination of an artificial neural network and sequential myoglobin measurements*, Proceedings of the Emergency Medicine Research Society Meeting, Cambridge

McGonigal M.D., Cole J., Schwab C.W., Kauder D.R., Rotondo M.F. & Angood P.B. (1993), *A new approach to probability of survival scoring for trauma quality assurance*. Journal of Trauma, 34(6):863–8

Ministry of Health (1997), *Annual Report*, Kuala Lumpur

Minsky, M. L. & Papert, S. A. (1988), *Perceptrons*, Expanded Edition, MA: MIT Press, Original edition, 1969

National Heart Attack Alert Program Working Group (1997), *An Evaluation of Technology for Identifying Acute Cardiac Ischemia in the Emergence Department*, Ann Emerg Med, 29,13-87

National Heart, Lung, and Blood Institute (2000), [www document]. URL. http://www.nhlbi.nih.gov/health/public/heart/other/hrtfail.htm

North, G. (1987), *A Celebration of Connectionism*, Nature, 328, 107

Ohno-Machado L. & Bialek D. (1998), *Diagnosing Breast Cancer from FNAs: Variable Relevance in Neural Network and Logistic Regression Models*. Brigham and Women's Hospital and Harvard Medical School

Open University (1985), *The Biology of Health and Disease*, Milton Keynes: Open University Press

Ortiz J., Ghefter C.G., Silva C.E. & Sabbatini R.M. (1995), *One-year mortality prognosis in heart failure: a neural network approach based on echocardiographic data*, J Am Coll Cardiol Dec;26(7):1586-93

Parker, D. (1985), *Learning Logic*: Technical Report TR-87, MA: MIT Press

Percy D.F. & Hine T.J. (1993), *Multivariate analysis of cholesterol distribution for monitoring the risk of coronary heart disease*, Statistics in Medicine, 12(10):967–74

Poggio, T., Torre, V. & Koch, C. (1985), *Computational Vision and Regularization Theory*, Nature, 31-319

Proyor, D.B., Shaw, L., McCants, C.B., Lee K.L., Mark, D.B. & Harrell F.E. (1993), *Value of the History and Physical in Identifying patients at Increased Risk of Coronary Artery Disease*, Ann Int Med, 118:81-90

Reid, D.D., Hamilton, P.J.S., McCartney, P., Rose, G., Jarrett, R.J. & Keen, H. (1976), *Smoking and Other Risk Factors for CHD in British Civil Servants*, Lancet, ii, 979-84

Rosenblatt, F. (1959), *Two Theorems of Statistical Separability in the Perceptron, Mechanization of Thought Processes*: Proceeding of a Symposium Held at the National Physical Laboratory, November 1958, 421-456

Rumelheart, D. E., Hinton, G. E. & Williams R. J. (1986), *Learning Internal Representation by Error Propagation*, Parallel Distributed Processing, 1: 675-695

Sharper, A.G., Pocock, S.J., Walker, M., Philips, A.W., Whitehead, T.P. & Macfarlane, P.W. (1985), *Risk factors for ischeamic heart disease: The Prospective Phase of the British Regional Heart Study*, Journal of Epidemiology and Community health, 39, 197-209

Shortliffe, E.H. (1976) *Computer-Based Medical Consultation: MYCIN,* American Elsevier, New York

Simkus, R. (1994), *Using Neural Networks In The Early Diagnosis Of Acute Myocardial Infarctions*, ITCH-94, Information Technology in Community Health Conference

Somoza, E. & Somoza, J.R. (1993), *A neural-network approach to predicting admission decisions in a psychiatric emergency room*, Medical Decision Making, 13(4):273–80

Stensmo, M., & Sejnowski, T.J. (1994), *A Mixture Model System for Medical and Machine Diagnosis*, Tech. Rep., INC-9401, Institute for Neural Computation, University of California, San Diego

Swales, J.D. (1981), *Clinical Hypertension*, London, Chapman & Hall

Truswell, A.S. (1985), *Diet and Hypertension*, British Medical Journal, 291, 125-7

Tsoukalas, L.H. & Uhrig, R.E. (1997), *Fuzzy and Neural Approaches in Engineering*, New York: John Wileys & Sons

Tu J.V. & Guerriere M.R. (1993), *Use of a neural network as a predictive instrument for length of stay in the intensive care unit following cardiac surgery*, Computers and Biomedical Research, 26(3):220–9

Werbos, P. (1974), *Beyond Regression: New Tools For Prediction and Analysis in the Behavioral Sciences* (Ph.D. Thesis), MA: Harvard U. Committee on Applied Mathematics

Appendix A Explanation on Symbols Used in Main Texts

The meaning of symbols that were used in the main text are as follows:

x_i	Input training vector
t_k	Output target vector
y_k	Output unit k
z_in_j	Weighted input signals for hidden unit j
y_in_k	Weighted input signals for output unit k
v_{ij}	Connection weights between input unit i and hidden unit j
w_{jk}	Connection weights between hidden unit j and output unit k
$f(z\_in_j)$, z_j	Activation function performed on weighted input signals for hidden unit j
$f(y\_in_k)$, y_k	Activation function performed on weighted input signals for output unit k
δ_k	Portion of error correction weight adjustment for w_{jk} that is due to an error at output unit y_k; also, the information about the error at unit y_k that is propagated back to the hidden units that feed into unit y_k
δ_j	Portion of error correction weight adjustment for v_{ij} that is due to back propagation of error information from the output layer to the hidden unit Z_j
α	Learning rate
Δw_{jk}	Adjustment of connection weights between hidden unit j and output unit k
Δv_{ij}	Adjustment of connection weights between input unit i and hidden unit j
$f^1(y\_in_k)$	Derivation of weighted input signals for output unit k
$f^1(z\_in_j)$	Derivation of weighted input signals for hidden unit j

Appendix B **Questionnaire**

MYOCARDIAL INFARCTION REGISTRY (ALOR SETAR HOSPITAL 2000)

PATIENT'S INFORMATION:

1. FULL NAME (ACCORDING TO IC):

2. PHONE NO:

3. DOA :

		/			/		

DOD

		/			/		

4. IDENTIFICATION CARD NO:
NEW:

						-			-				

OLD:

5. BIRTH DATE:

		/			/		

6. GENDER:

MALE	
FEMALE	

7. ETHNIC GROUP:

MALAY	
CHINESE	
INDIAN	
SIAMESE	
OTHERS	

IF OTHERS, PLEASE SPECIFY: ____________________

8. SITE OF MYOCARDIAL INFARCTION:

ANTERIOR/ANTEROSEPTAL/ANTEROLATERAL	
INFERIOR WITH OR WITHOUT POSTERIOR WALL OR RV INVOLVEMENT	

NONQ	
OTHERS	

IF OTHERS, PLEASE SPECIFY: ____________________

9. **RISK FACTORS:**

SMOKER	
HYPERTENSION	
DIABETES MELLITUS	
FAMILY HISTORY OF PREMATURE CAD	
OTHERS	

IF OTHERS, PLEASE SPECIFY: ____________________

10. **INVESTIGATION (ON ADMISSION):**

TOTAL COLESTEROL		MMOL/L
TRIGLYCERIDES		MMOL/L
BLOOD GLUCOSE		MMOL/L
UREA		MMOL/L
CREATININE		UMOL/L
PEAK CREATINE KINASE		U/L

11. **COMPLICATION OF MYOCARDIAL INFARCTION**

HEART FAILURE/PULMONARY OEDEMA REQUIRING VENTILATION	
BRADYARRHYTMIAS REQUIRING PACING	
TACHYARRHYTHMIAS REQUIRING CARDIOVERSION OR ANTIARRHYTHMICS	
CARDIOGENIC SHOCK	
DEATH	
OTHERS:	

IF OTHERS, PLEASE SPECIFY: ____________________

12. **MANAGEMENT :**

	YES	NO
STREPTOKINASE GIVEN :		

TIME DELAY TO THROMBOLYSIS : ________HOURS ______MINS

ACEI	
ASPIRIN	

BETA-BLOKER	
CALCIUM ANTAGONISTS	
NITRATE	
STATINS	
OTHERS	

IF OTHERS, PLEASE SPECIFY: ______________________

13. RISK STRATIFICATION:

A: 2D ECHOCARDIOGRAPHY:

DATE: ____________

LV EJECTION FRACTIONS:

NORMAL(50% OR MORE)	
BETWEEN 35-49%	
BETWEEN 20-34%	
LESS THAN 20%	
NOT DONE	

B: EXERCISE STRESS TEST:

DATE: ____________

TEST RESULT: ______________________

TOTAL EXERCISE TIME (IN MINS)	
TOTAL METS ACHIEVED	
SIGNIFICANT ST-T CHANGES	
ANGINA	
ARRHYTHMIAS	
CONCLUSION	

C: REFERRAL FOR ANGIOGRAPHY:

YES		NO	

14. SURVIVE

YES		NO	

Appendix C Trained Neural Network Weights File

"Number of Input ",6
"Number of Hidden ",8
"Weight of input-hidden ",0,0,-8.38491259961198E-02
"Weight of input-hidden ",0,1,.010875806982158
"Weight of input-hidden ",0,2,-.252100775654427
"Weight of input-hidden ",0,3,.567932639547908
"Weight of input-hidden ",0,4,-.306315643054388
"Weight of input-hidden ",0,5,1.0467886810617
"Weight of input-hidden ",0,6,-.154465249640084
"Weight of input-hidden ",0,7,-.512491852626331
"Weight of input-hidden ",1,0,-.808362141350082
"Weight of input-hidden ",1,1,-.291145660888625
"Weight of input-hidden ",1,2,.525055236349283
"Weight of input-hidden ",1,3,-.044429258773935
"Weight of input-hidden ",1,4,-.677006224896843
"Weight of input-hidden ",1,5,1.66360849199704
"Weight of input-hidden ",1,6,-.813114460958249
"Weight of input-hidden ",1,7,-.919678322262907
"Weight of input-hidden ",2,0,-.629546273836682
"Weight of input-hidden ",2,1,-.14919830242663
"Weight of input-hidden ",2,2,.242698824190468
"Weight of input-hidden ",2,3,-.139540725013938
"Weight of input-hidden ",2,4,-.76988287594639
"Weight of input-hidden ",2,5,1.4951041205113
"Weight of input-hidden ",2,6,-1.5308875174703
"Weight of input-hidden ",2,7,-.386639323130543
"Weight of input-hidden ",3,0,-.253749254349766
"Weight of input-hidden ",3,1,-.805177794399322
"Weight of input-hidden ",3,2,.622376617546733
"Weight of input-hidden ",3,3,-.902948375034292
"Weight of input-hidden ",3,4,-.10821668197757
"Weight of input-hidden ",3,5,-.795983170278344
"Weight of input-hidden ",3,6,.372868341698578
"Weight of input-hidden ",3,7,-.257339704405625
"Weight of input-hidden ",4,0,.266952876582998
"Weight of input-hidden ",4,1,-.151408340119139
"Weight of input-hidden ",4,2,-.233742738363257
"Weight of input-hidden ",4,3,-.219291178355374
"Weight of input-hidden ",4,4,4.70314866853041E-02
"Weight of input-hidden ",4,5,-.255624919949496
"Weight of input-hidden ",4,6,.539776889820466
"Weight of input-hidden ",4,7,-.17699791180853
"Weight of input-hidden ",5,0,-1.51153154170376

"Weight of input-hidden ",5,1,-.984888605992233
"Weight of input-hidden ",5,2,1.22105161882499
"Weight of input-hidden ",5,3,-1.31810496257749
"Weight of input-hidden ",5,4,-1.20346756798088
"Weight of input-hidden ",5,5,1.18831084814797
"Weight of input-hidden ",5,6,-1.95772981425357
"Weight of input-hidden ",5,7,-.95483525988469
"Weight of hidden-output ",0,-2.96534296086597
"Weight of hidden-output ",1,-1.93396602068863
"Weight of hidden-output ",2,1.67159698163377
"Weight of hidden-output ",3,-2.28201041241667
"Weight of hidden-output ",4,-2.25261136916391
"Weight of hidden-output ",5,2.84115214430523
"Weight of hidden-output ",6,-3.61801630231659
"Weight of hidden-output ",7,-2.0876571484759

Appendix D Sample Raw Data

Smoker	HyperTension	Diabetes	Family His of CAD	Cholesterol	Age	Angina
YES	YES	NO	YES	4.7	50	NO
YES	NO	YES	YES	0	50	YES
YES	NO	NO	NO	5.6	48	NO
YES	NO	YES	NO	6.6	50	YES
NO	NO	NO	NO	3.4	49	NO
NO	YES	NO	NO	6.1	46	NO
NO	NO	NO	NO	5.1	33	NO
YES	NO	NO	YES	7.6	41	NO
YES	NO	NO	NO	0	76	NO
YES	YES	YES	NO	5.4	58	NO
YES	YES	YES	NO	0	70	NO
NO	YES	NO	NO	6.1	46	NO
YES	NO	NO	NO	4.86	76	NO
NO	NO	NO	NO	5.47	81	NO
YES	YES	NO	NO	0	79	NO
YES	YES	NO	NO	5.8	62	NO
YES	YES	NO	NO	5.3	69	NO
YES	NO	YES	YES	7.8	41	NO
YES	YES	YES	NO	7.3	47	NO
YES	NO	NO	NO	0	63	NO
YES	YES	NO	NO	0	75	NO
YES	YES	YES	NO	0	65	NO
YES	NO	YES	NO	0	47	NO
YES	NO	NO	NO	0	48	NO
NO	NO	NO	NO	0	66	NO
YES	NO	NO	NO	6.44	58	NO
YES	NO	NO	YES	6.28	29	NO
YES	NO	NO	NO	0	55	NO
YES	NO	NO	NO	0	59	NO
YES	NO	NO	NO	4.3	30	NO
NO	NO	NO	NO	3.5	67	NO
YES	YES	NO	NO	5.6	46	NO
YES	NO	NO	NO	6.6	60	NO
YES	NO	NO	NO	6.3	44	NO
YES	NO	NO	NO	5.7	51	NO
NO	NO	NO	NO	8.8	62	NO

Appendix E Sample Preprocess Data

Smoker	HyperTension	Diabetes	Family His of CAD	Cholesterol	Age	Angina
1	1	0	1	0.580246914	0.617283951	1
1	0	1	1	0.725308642	0.617283951	1
1	0	0	0	0.691358025	0.592592593	0
1	0	1	0	0.814814815	0.617283951	1
0	0	0	0	0.419753086	0.604938272	0
0	1	0	0	0.75308642	0.567901235	1
0	0	0	0	0.62962963	0.407407407	0
1	0	0	1	0.938271605	0.50617284	0
1	0	0	0	0.725308642	0.938271605	1
1	1	1	0	0.725308642	0.864197531	1
1	1	1	0	0.725308642	0.802469136	1
1	0	1	0	0.725308642	0.580246914	0
1	0	1	1	0.962962963	0.50617284	1
1	0	0	0	0.725308642	0.592592593	0
0	0	0	0	0.725308642	0.814814815	0
1	0	0	0	0.725308642	0.777777778	1
1	0	0	1	0.775308642	0.358024691	0
1	1	0	0	0.725308642	0.975308642	1
1	0	0	0	0.725308642	0.728395062	0
1	0	0	0	0.530864198	0.37037037	0
1	1	1	0	0.666666667	0.716049383	1
1	1	0	0	0.691358025	0.567901235	0
1	0	0	0	0.814814815	0.740740741	0
1	0	0	0	0.777777778	0.543209877	1
1	0	0	0	0.703703704	0.62962963	0
1	1	1	0	1	0.592592593	1
1	0	0	0	0.725308642	0.49382716	0
1	0	0	0	0.725308642	0.555555556	0
1	1	1	0	0.851851852	0.592592593	1
1	0	0	0	0.728395062	0.592592593	1

Appendix F User Manual

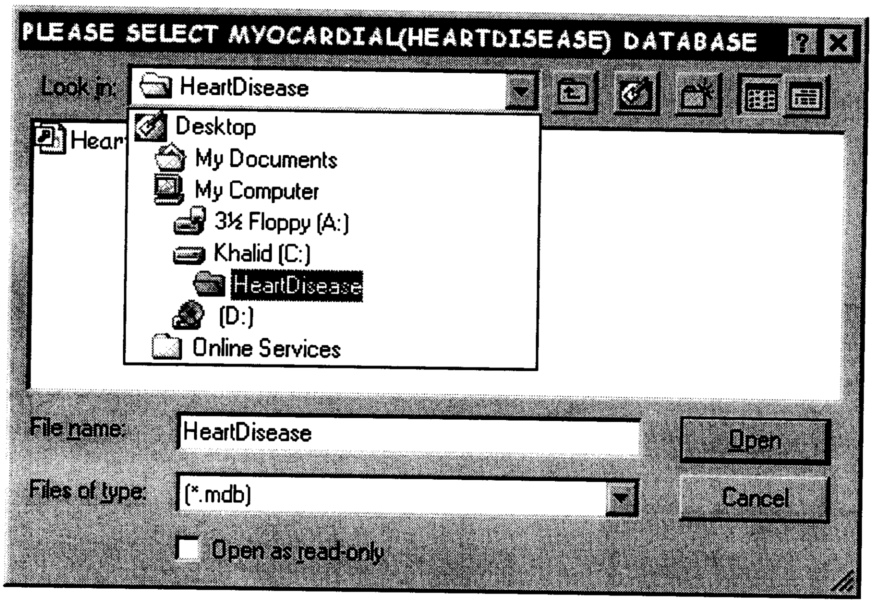
TABLE OF CONTENT

1.0 Flash Screen

- This is the introductory screen.
- In order to proceed, click your mouse or Wait 5 seconds.

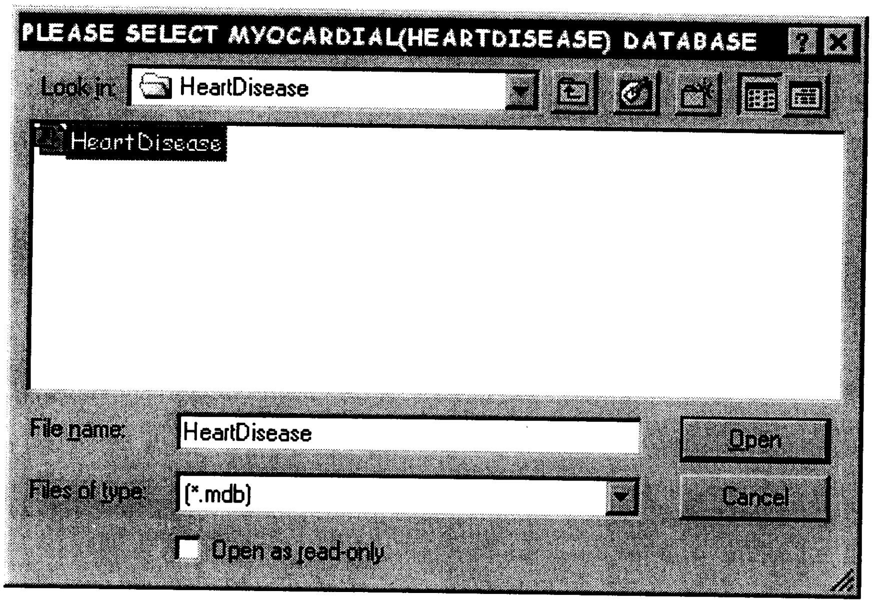
2.0 Database Selection

- Select database location and database name (Database location is equivalent to the location selected during the Installation ie. 'C:\HeartDisease' .



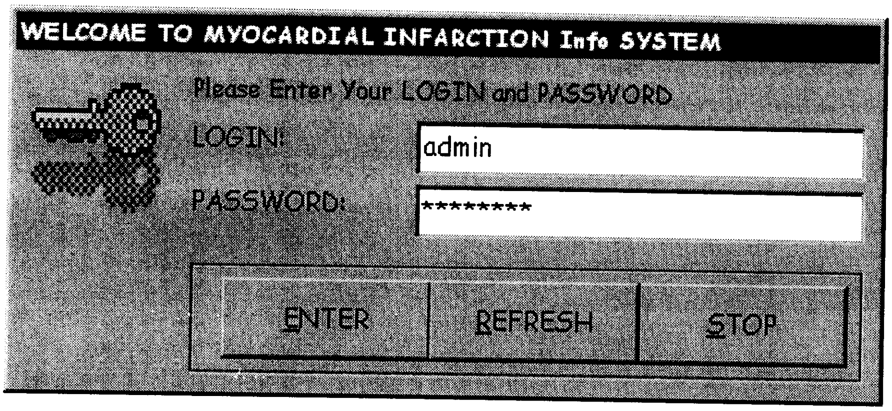
3.0 Filename Selection

- Select database name ie. 'HeartDisease' and click OK to proceed.



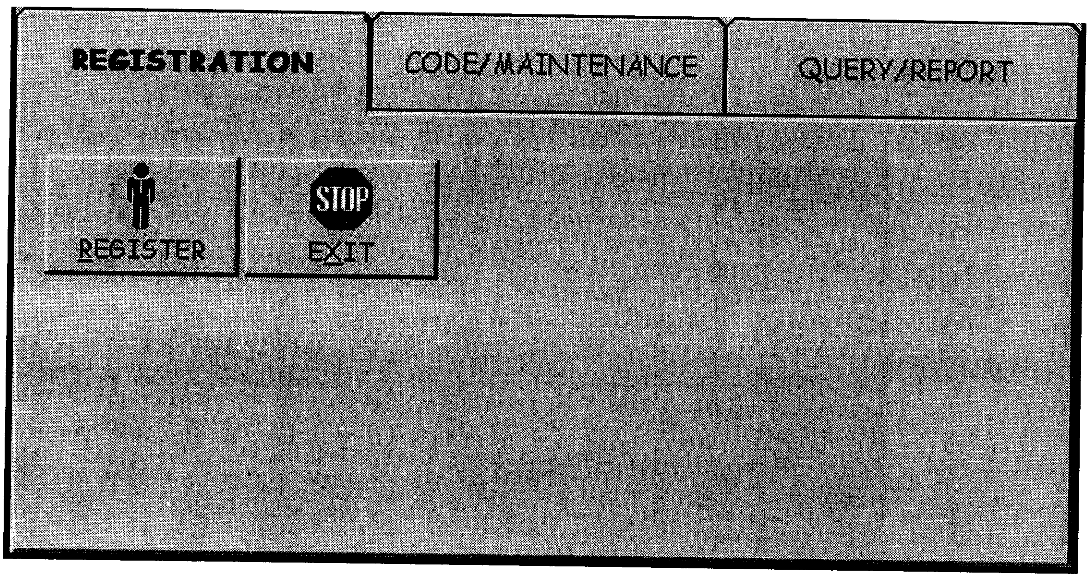
4.0 Username and Password

- Enter 'admin' if you are the system administrator, else enter your login name such as 'khalid' and followed by password.
- System Administrator will have a full access to the MIMIS (including the NNS Training and User Maintenance)
- Then Hit enter to proceed.



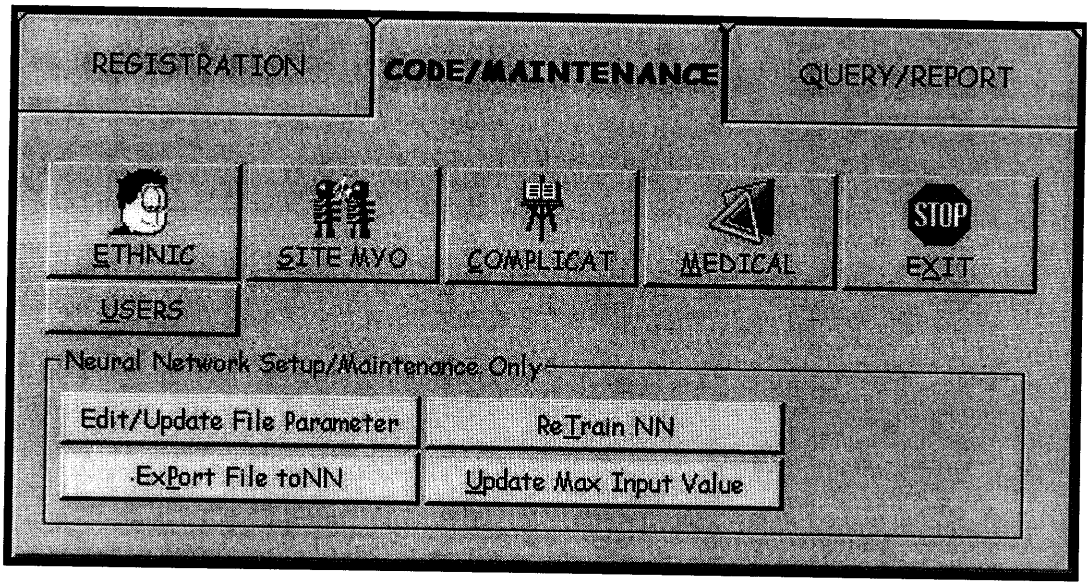
5.0 Main Menu for Registration

- Use to register patient upon addmision
- Use to edit/update patient record

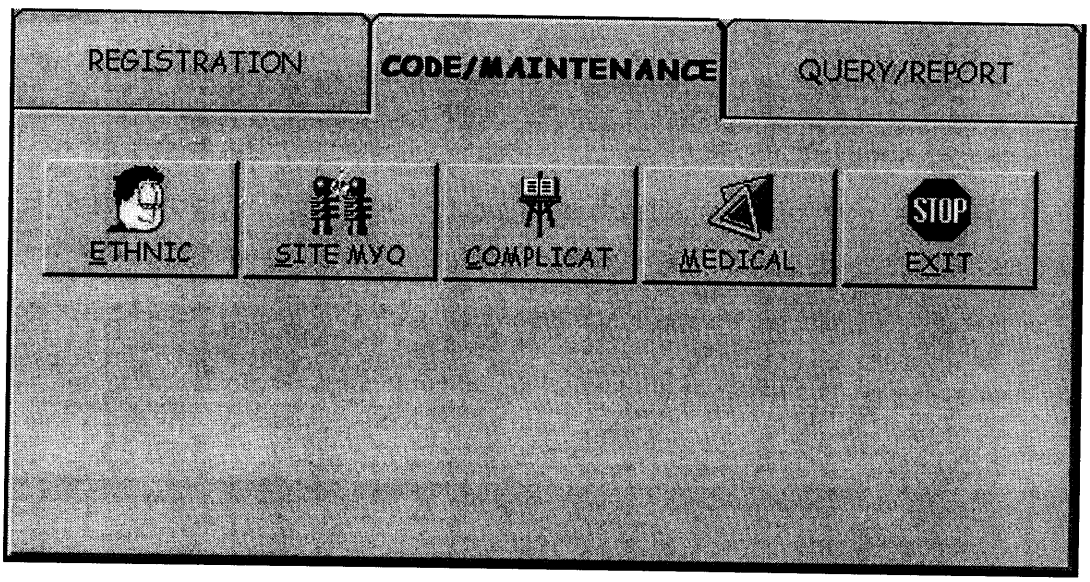


6.0 Main Menu for Code and Maintenance (admin only)

- Use to manipulate code for
  - Ethnic
  - Site of Myo Infarc
  - Complication of Myo Infarc
  - Medical Info
- If you login as an **admin**, you also can access
  - User Maintenace
  - Neural Network Simulator

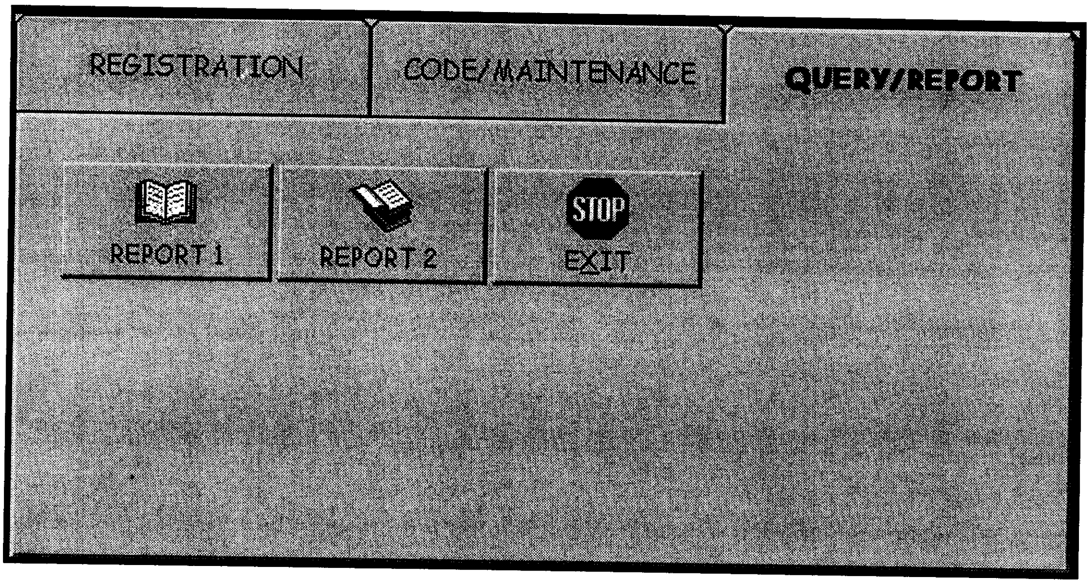


7.0 Main Menu for Code and Maintenance (normal user)



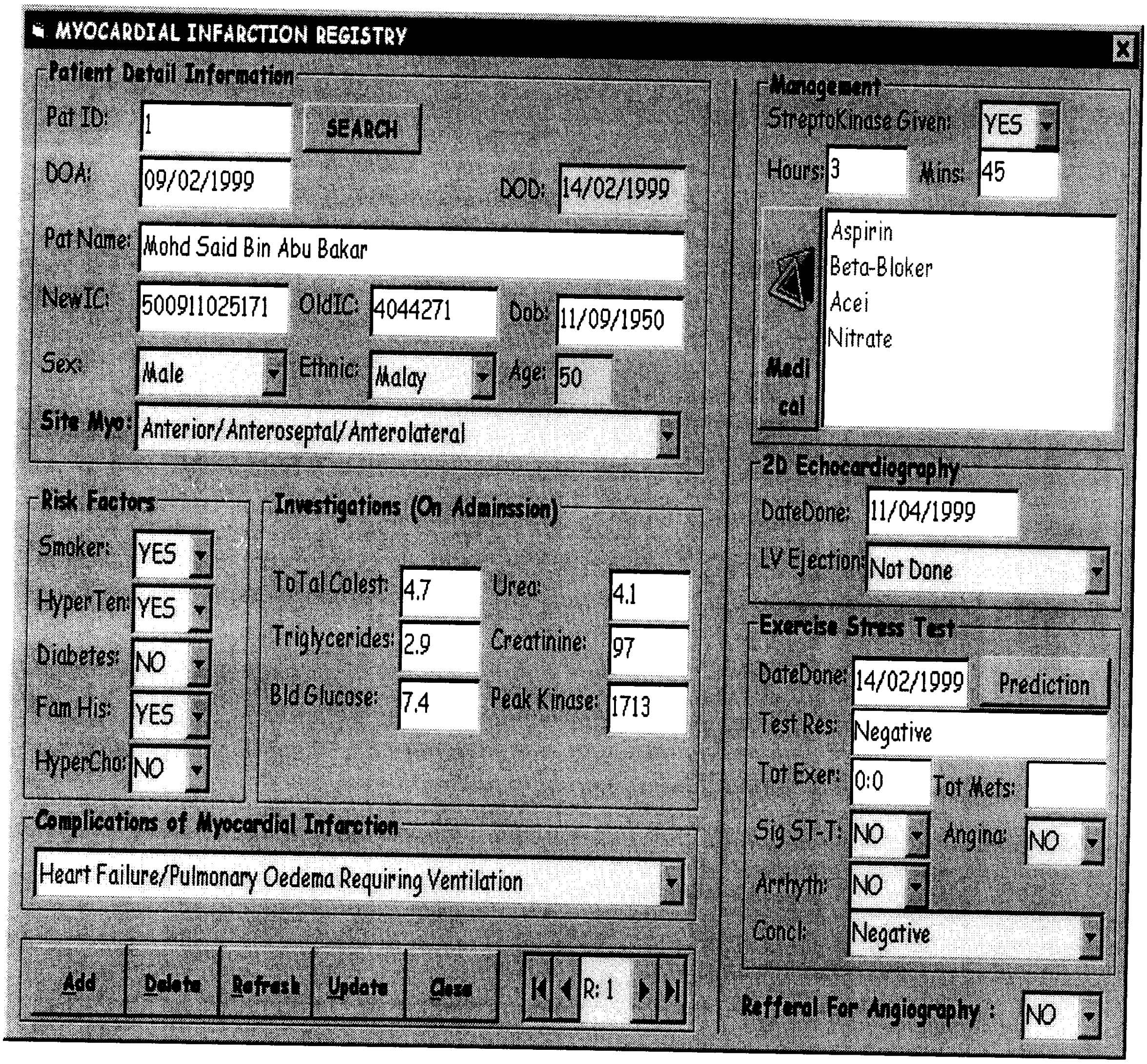
8.0 Main menu for Query and Report

✍ Use to View and Print Report



9.0 Registration Menu

- Use to Add new patient
- Use to Search particular patient
- Use to Delete existing patient
- Use to Edit/Update patient record



9.1 Add new patient

9.1.1 Patient Detail information

A. Click Add Button
   - Pat ID is automatically generated (by adding 1 to the last Pat ID)

B. Enter DOA (date of Admission) in DD/MM/YYYY

C. Enter Patient Name

D. Enter New IC without any space

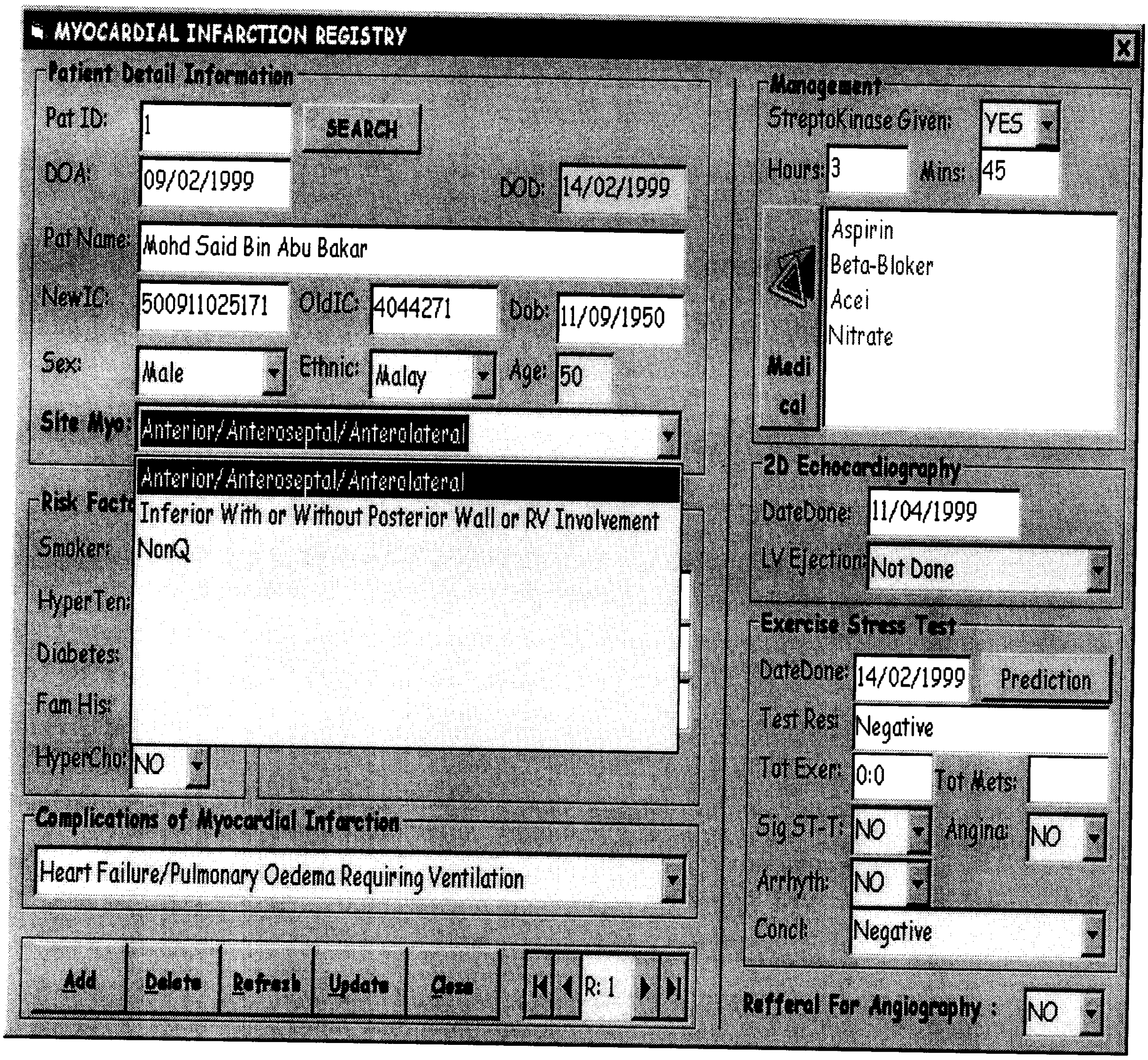
E. Enter Old IC

F. Enter Dob (Date of Birth) If NewIC is filled then it's auto generated

G. Enter Sex by clicking the ComboBox .If NewIC is filled then it's auto generated

H. Enter Ethnic by clicking the ComboBox
   - If Ethnic is not stated in the ComboBox, double-click the ComboBox to add new one.

I. Enter Site of Myo Infarc by click the ComboBox

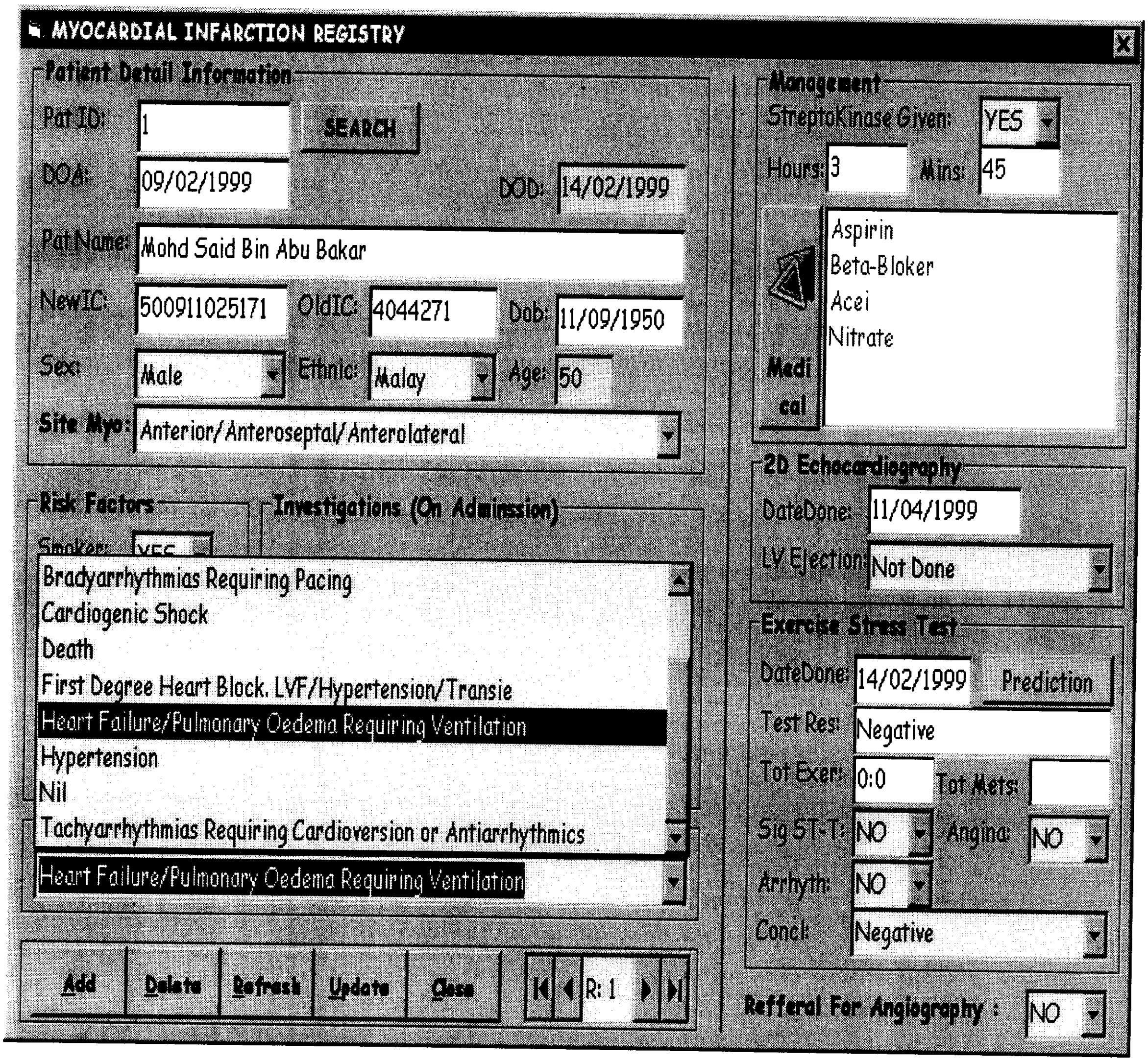


9.1.2 Risk Factors

A. Enter Smoking by clicking the ComboBox
B. Enter Hyper Tension by clicking the ComboBox
C. Enter Diabetes by clicking the ComboBox
D. Enter Family History of Premature CAD by clicking the ComboBox
E. Enter Hypercholesterolemia by clicking the ComboBox
F. Enter Total Cholesterol (number)
G. Enter Triglycerides (number)
H. Enter Blood Glucose
I. Enter Urea
J. Enter Creatinine
K. Enter Peak Creatine Kinase

9.1.3 Complication of Myocardial Infarction

- Enter Complication of Myo Infarc by clicking the ComboBox



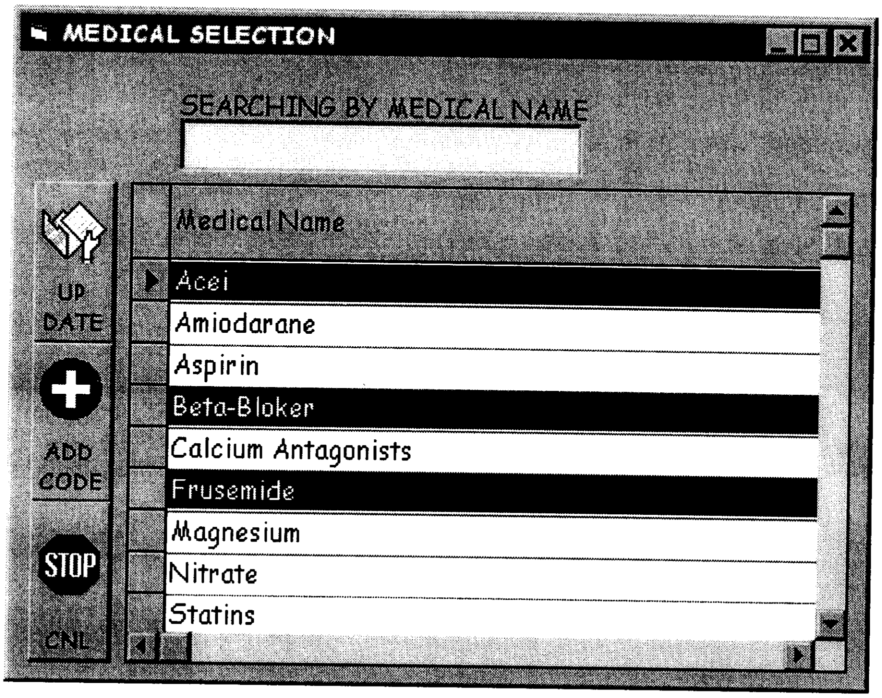
9.1.4 Management

A. Enter Streptokinase Given by clicking the ComboBox
B. Enter Time Delay to Thrombolysis in Hours
C. Enter Time Delay to Thrombolysis in Mins
D. Click on Medical to assign medicine

9.1.4.1 Medical Selection Form

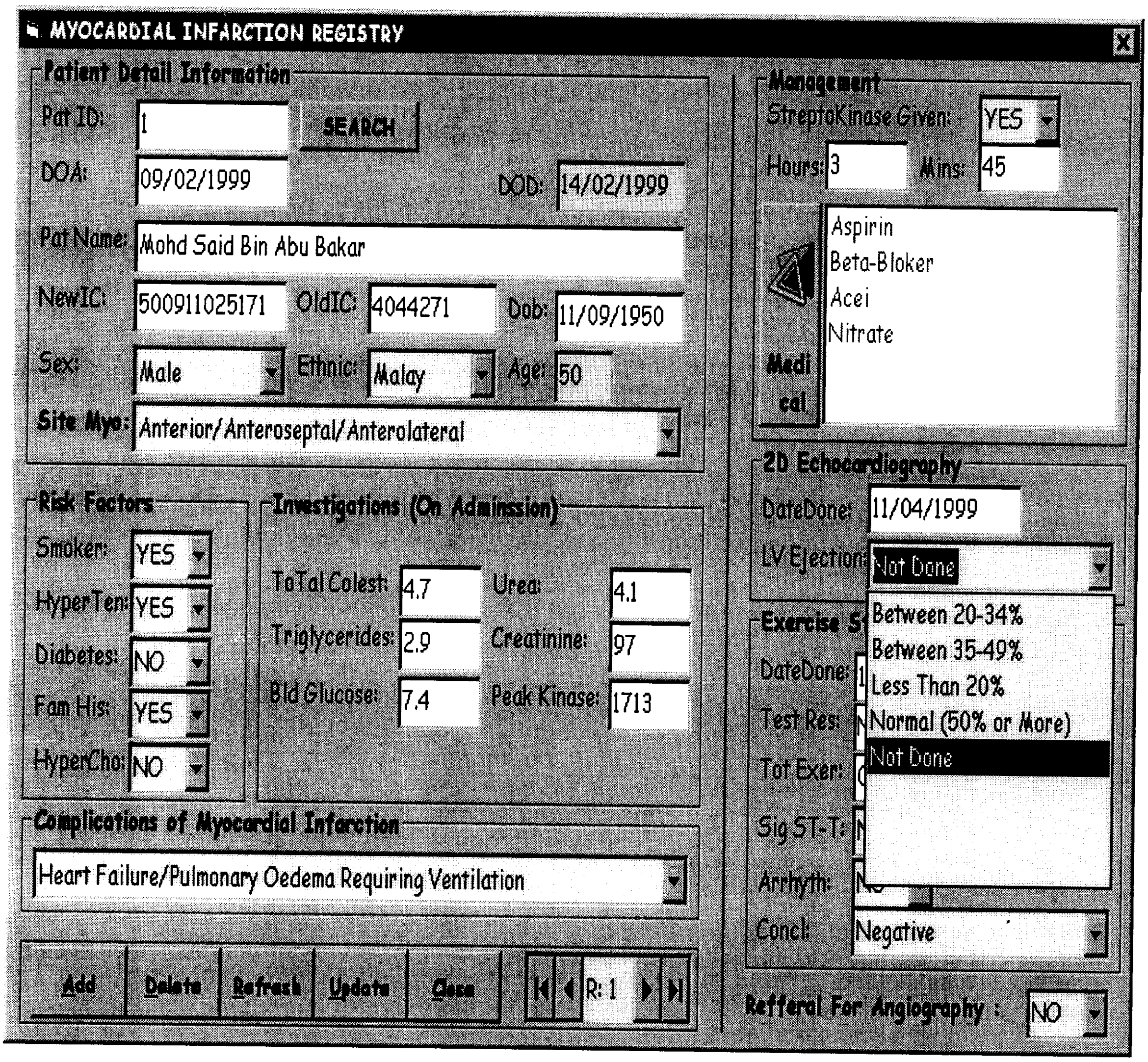
A. Select desired medical by clicking the Medical Name
   - To Select Multiple Medical, hold the <SHIFT> key and clicking any desired medical.
   - If desired medical is not in the list, click ADD CODE to add new one.

B. Click UPDATE button to confirm selection or CANCEL to cancel selection



9.1.5 2D Echocardiography

A. Enter DateDone in DD/MM/YYYY
B. Enter LV Ejection by clicking the ComboBox



9.1.6 Stress Test

A. Enter DateDone in DD/MM/YYYY
B. Enter TestResult
C. Enter Total Mets Achieved
D. Enter Significant ST-T Changes by clicking the ComboBox
E. Enter Angina by clicking the ComboBox
F. Enter Arrhythhmias by clicking the ComboBox
G. Enter Conclusion by clicking the ComboBox

9.1.7 Referral for Angiography

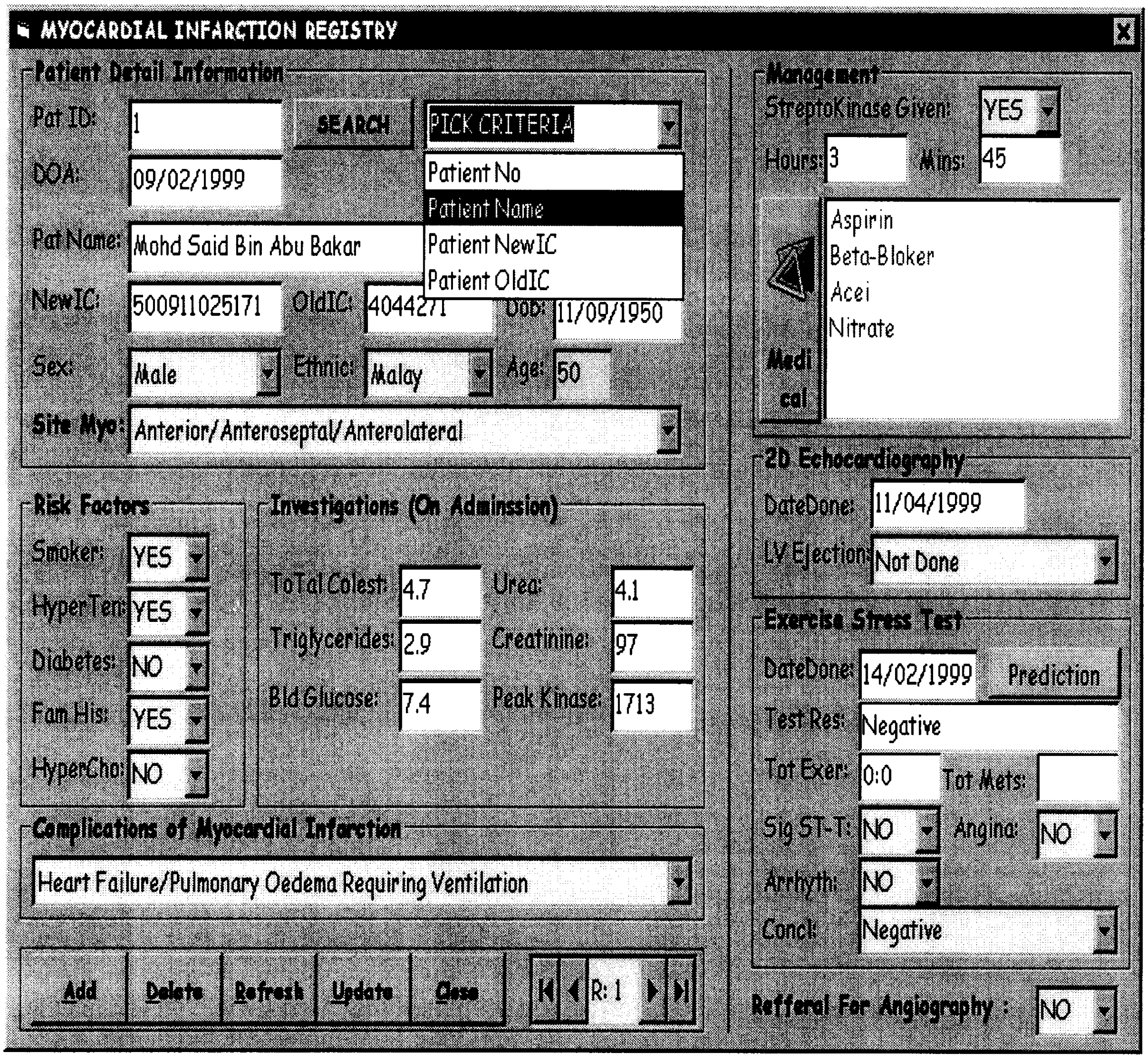
- Enter Refferal for Angiography by clicking the ComboBox

9.2 Search Particular Patient

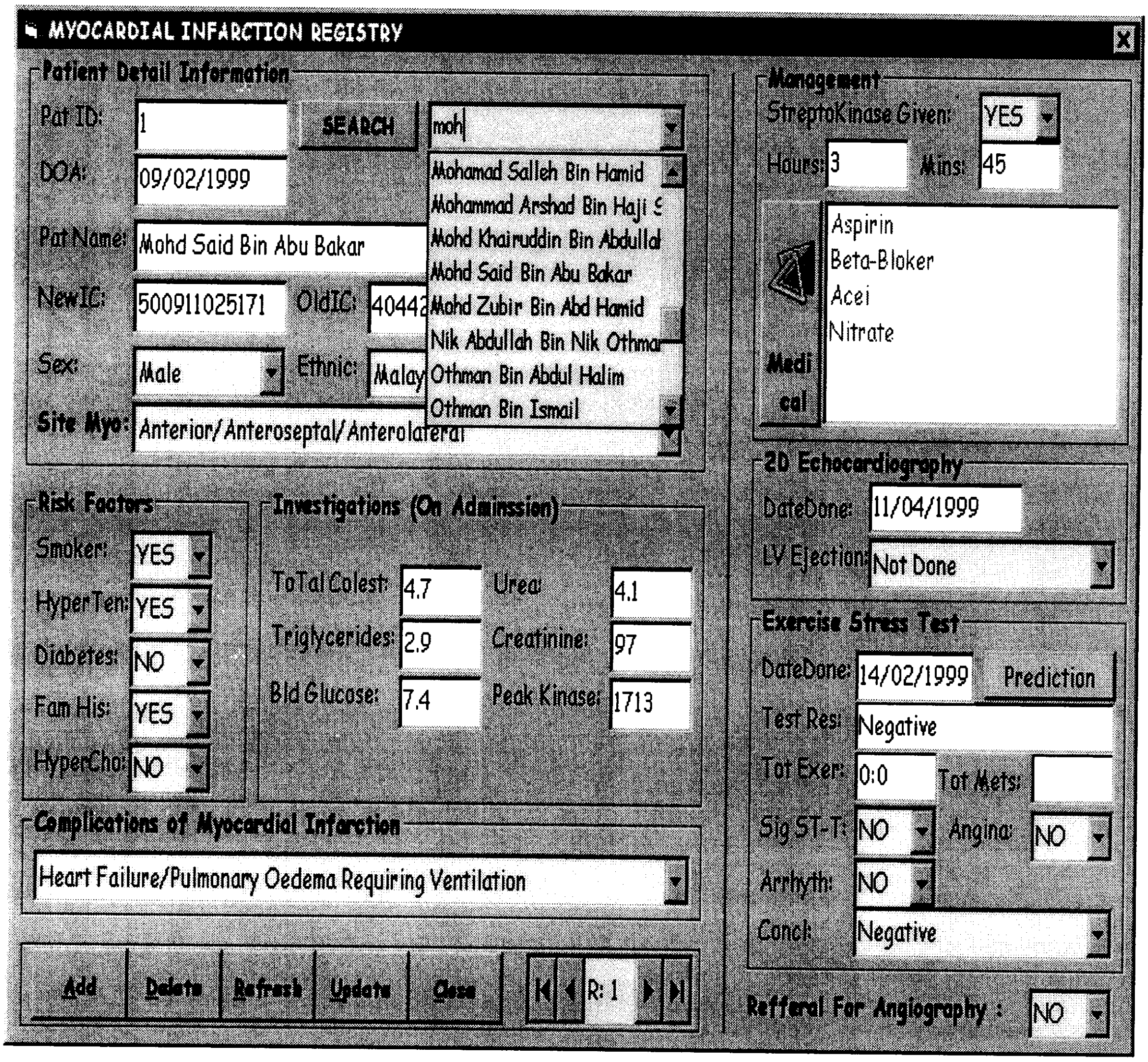
A. Click SEARCH button

B. ComboBox PICK CRITERIA will display

- Select Criteria ie. Patient Name to search patient by name



C. A list of Patient Name is display in ListBox
D. Select any desired patient
E. Selected record will be display



9.3 Delete Patient Record

A. Select desired record to Delete by using Search Particular Patient in 9.2
B. Delete the Record by clicking the DELETE button
C. A confirmation dialog will appear to confirm
   - Click YES to Delete or NO to Cancel

9.4 Edit/Update Patient Record

A. Select desired record to Edit sing Search Particular Patient in 9.2
B. Edit Record by changing desired one
C. Click Update to confirm the changes
D. A confirmation dialog will appear to confirm
   - Click YES to Delete or NO to Cancel

10.0 Maintenance Code Menu

- Add New Code
- Delete Code
- Edit/Update Code

10.1 Add New Code

A. Click ADD button
B. Enter Code
C. Enter Name
D. Click UPDATE Button

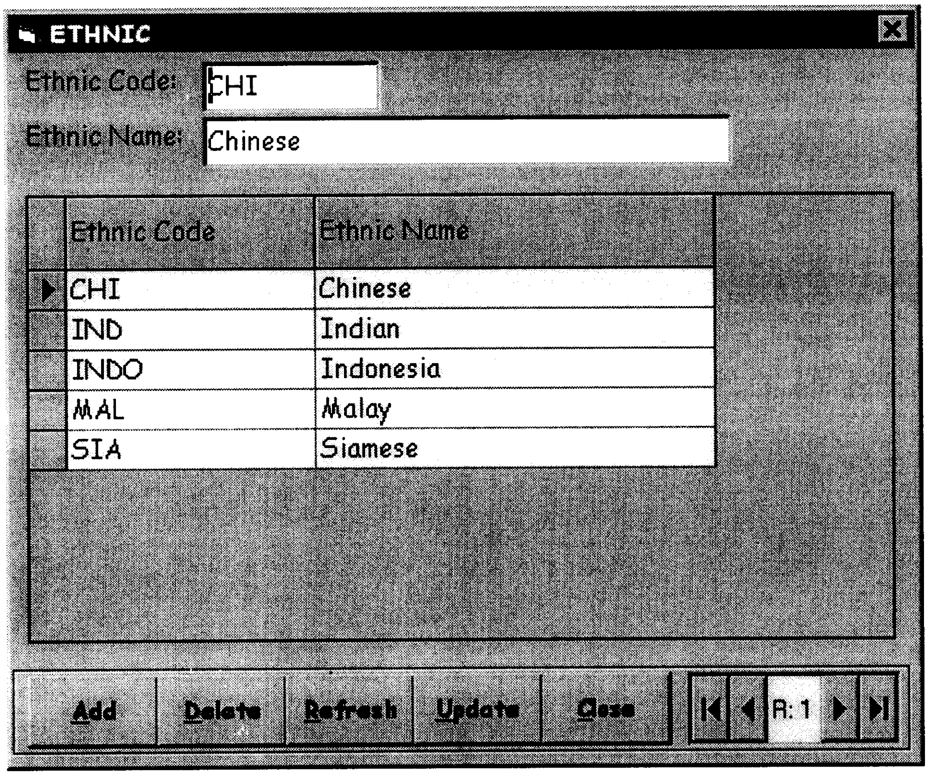
10.2 Delete Code

A. Select Desired Code to delete
B. Click DELETE button
C. Confirmation dialog will appear to confirm
   - Click YES to Delete
   - Click NO to Cancel

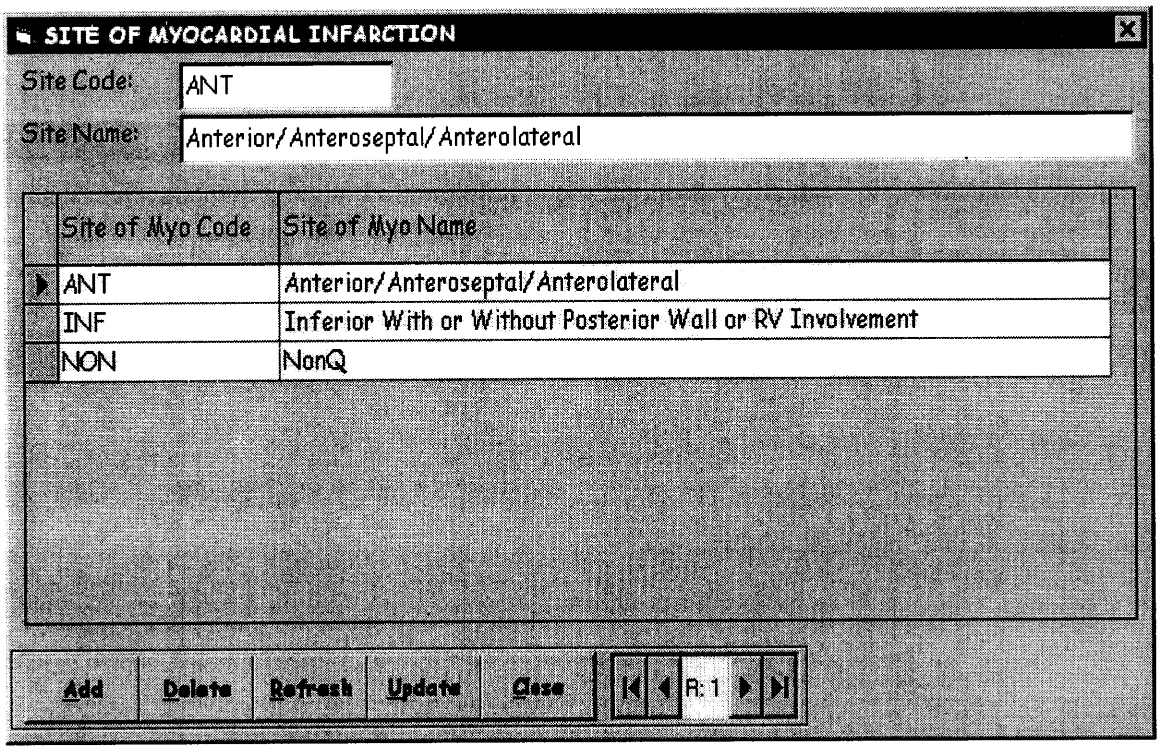
10.3 Edit/Update Code

A. Select Desired Code to edit
B. Make changes
C. Click Update button
D. Confirmation dialog will appear to confirm
   - Click YES to Delete
   - Click NO to Cancel

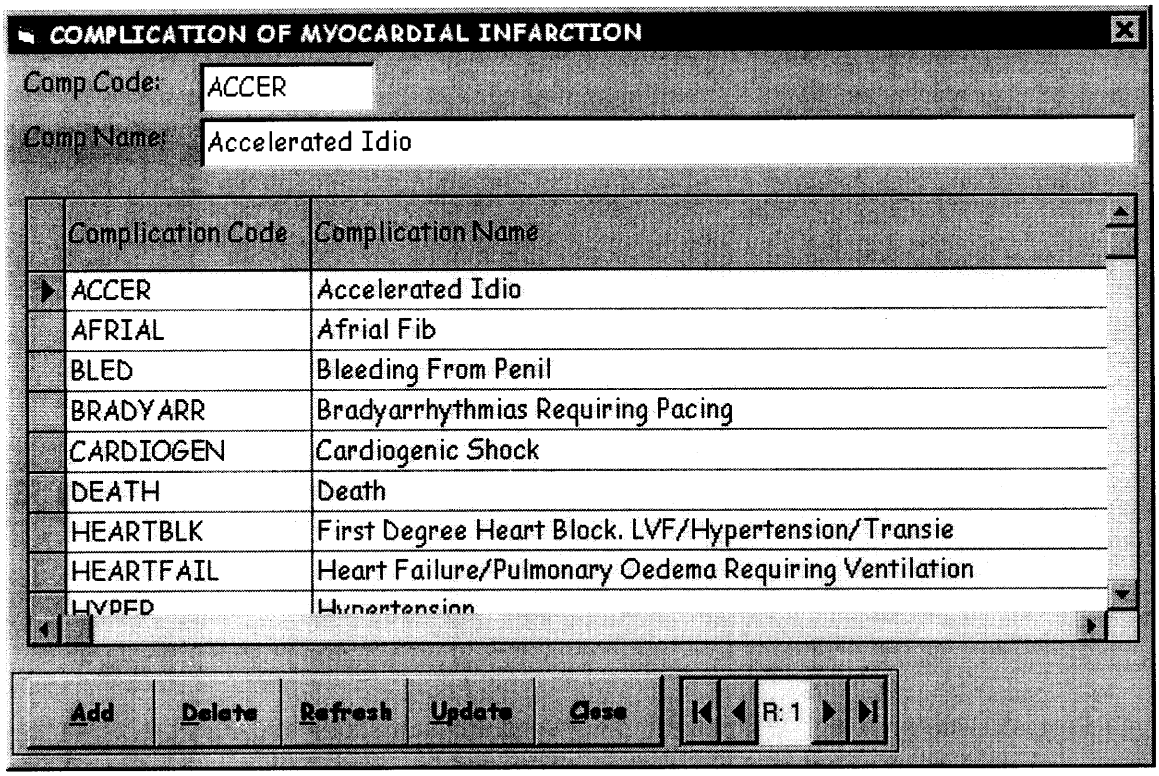
10.4 Ethnic Code



10.5 Site of Myocardial Infarction Code



10.6 Complication of Myocardial Infarction Code



COMPLICATION OF MYOCARDIAL INFARCTION

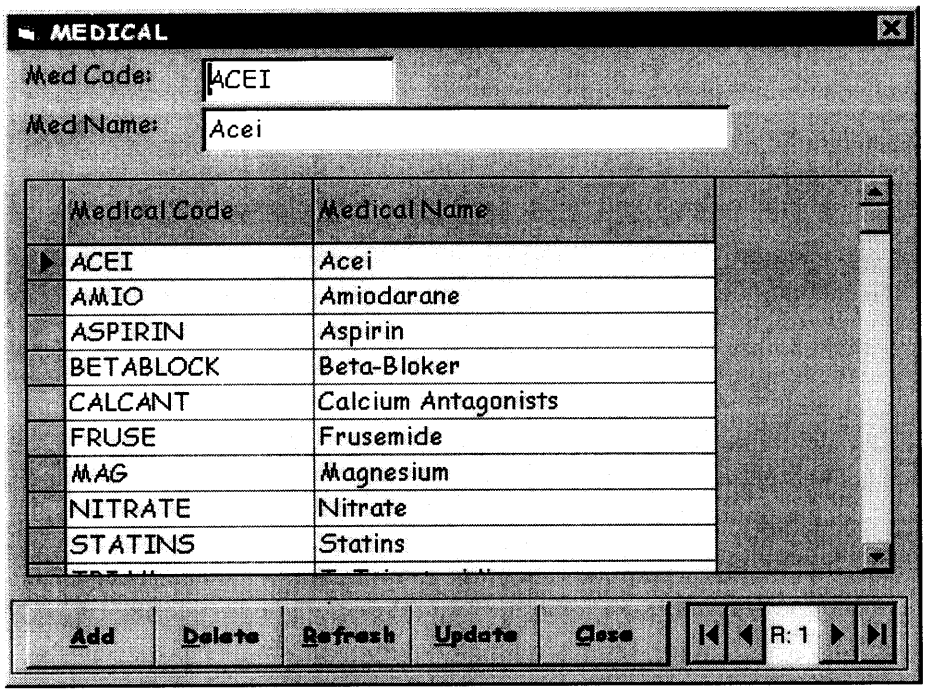
Comp Code: ACCER

Comp Name: Accelerated Idio

Complication Code	Complication Name
ACCER	Accelerated Idio
AFRIAL	Afrial Fib
BLED	Bleeding From Penil
BRADYARR	Bradyarrhythmias Requiring Pacing
CARDIOGEN	Cardiogenic Shock
DEATH	Death
HEARTBLK	First Degree Heart Block. LVF/Hypertension/Transie
HEARTFAIL	Heart Failure/Pulmonary Oedema Requiring Ventilation
HYPER	Hypertension

Add Delete Refresh Update Close R: 1

10.7 Medical Code



MEDICAL

Med Code: ACEI

Med Name: Acei

Medical Code	Medical Name
ACEI	Acei
AMIO	Amiodarane
ASPIRIN	Aspirin
BETABLOCK	Beta-Bloker
CALCANT	Calcium Antagonists
FRUSE	Frusemide
MAG	Magnesium
NITRATE	Nitrate
STATINS	Statins

Add Delete Refresh Update Close R: 1

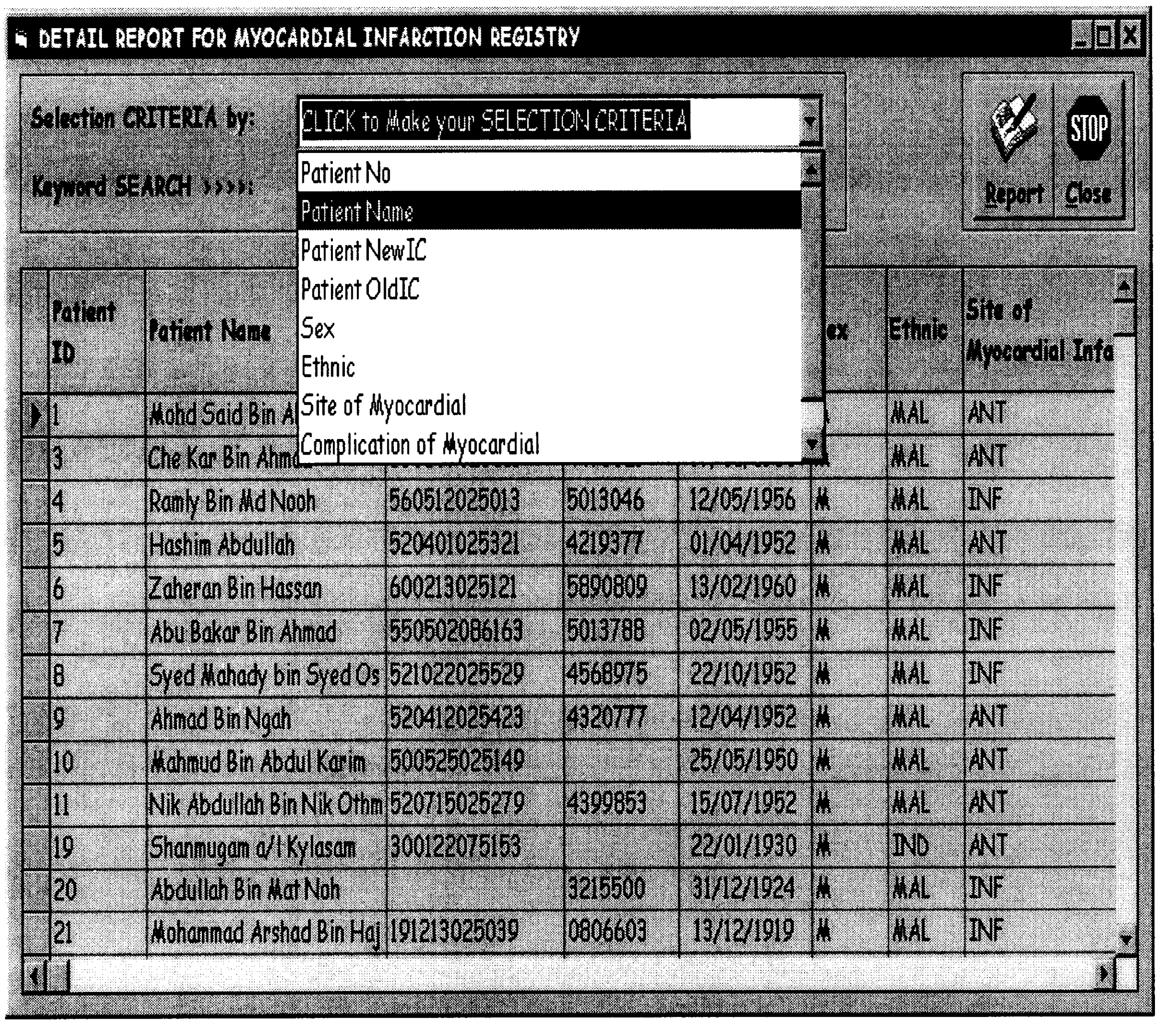
11.0 Report and Query Menu

- Report on Patient detail information
- Report on Medical information

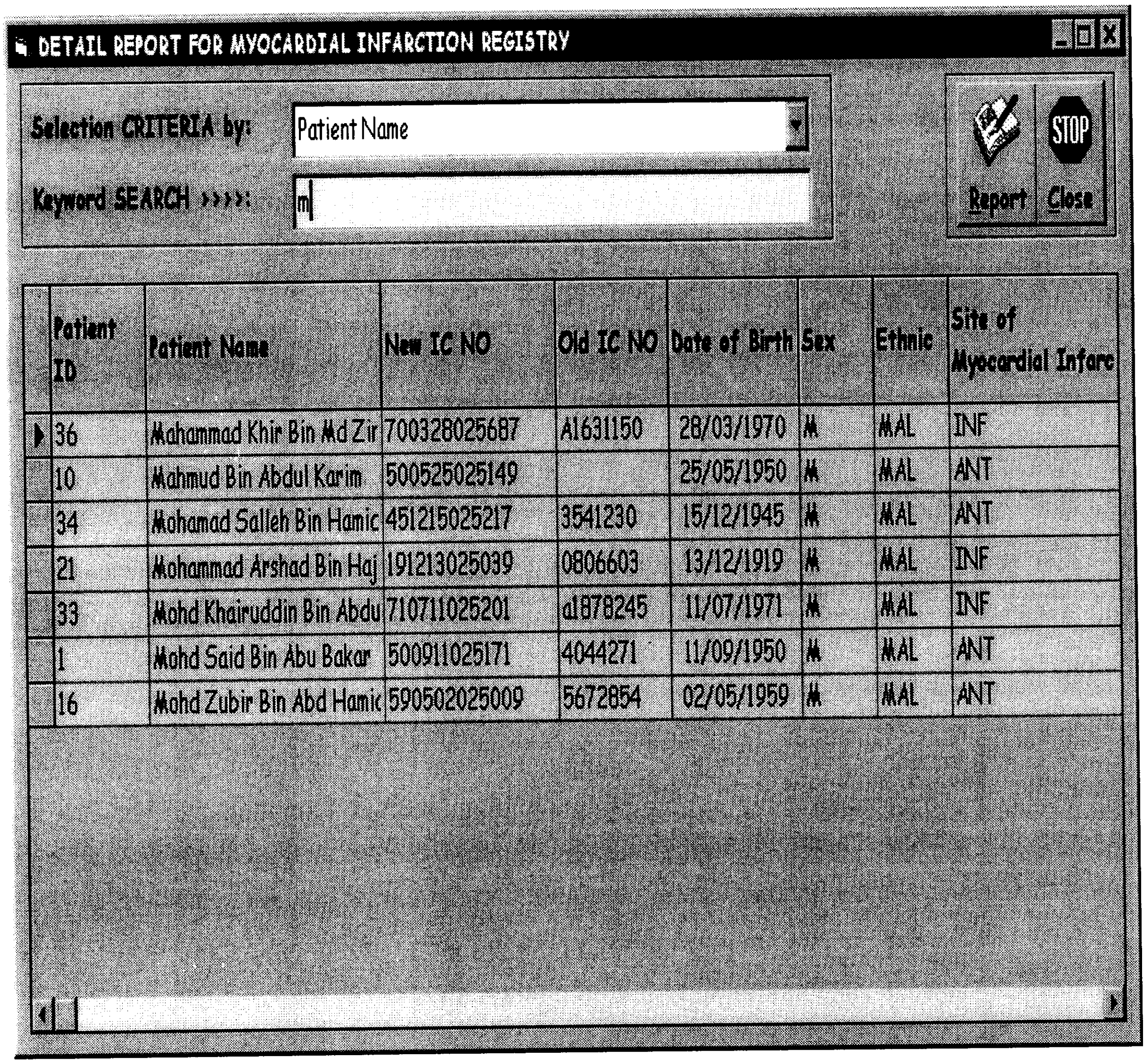
11.1 Query and Report Selection Criteria

Report can be view or printed based on the following selection:

- Patient No
- Patient Name
- Patient NewIC
- Patient OldIC
- Sex
- Ethnic
- Site of Myocardial Infarction
- Complication of Myocardial Infarction
- Referral for Angiography

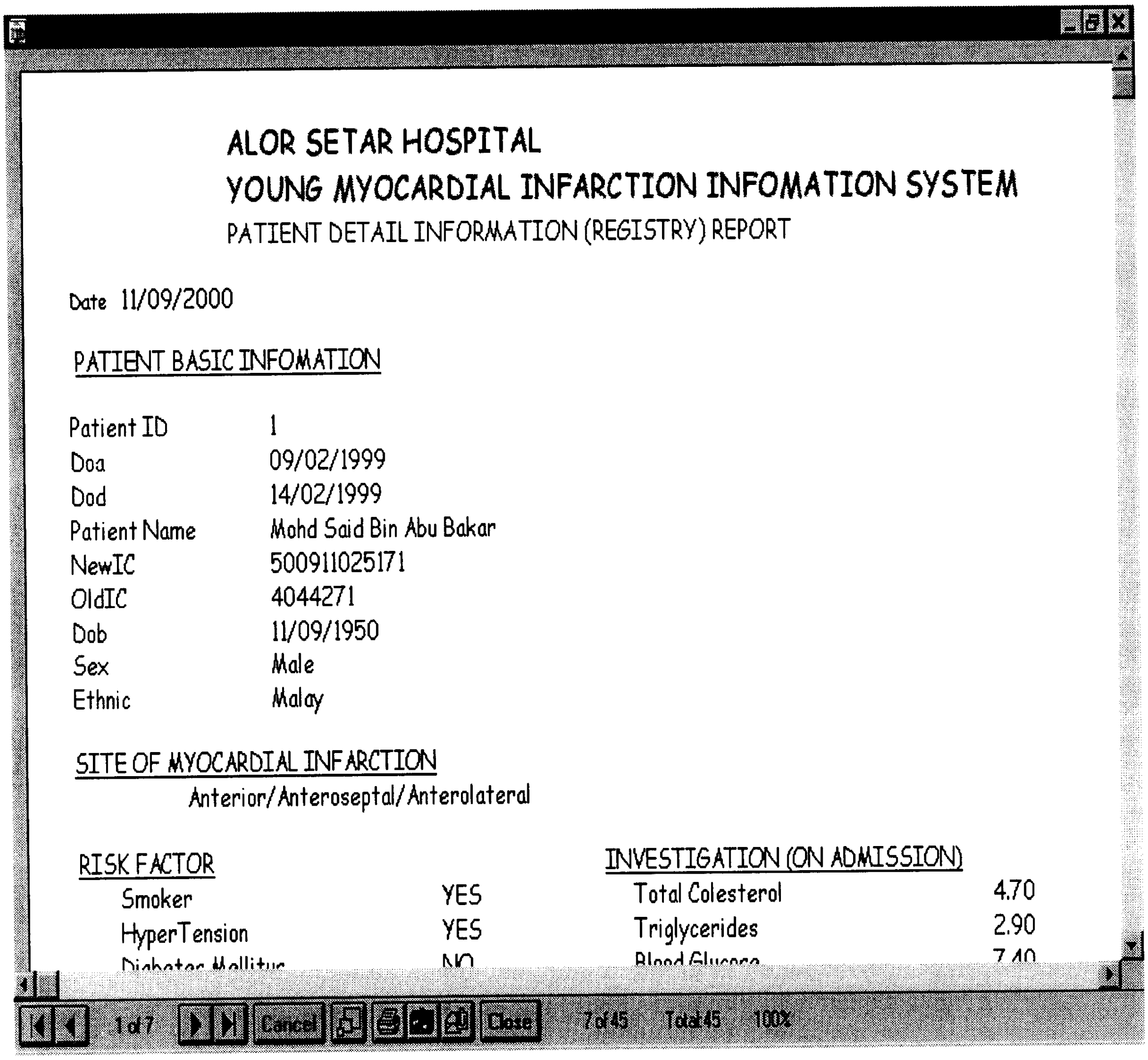


A. Select Selection Criteria by clicking the ComboBox
   - o Ie. Select the Name Criteria
B. Enter Keyword Search by Patient Name (ie : All Patient name starting with 'm')
C. All patient that begin with the letter 'm or M' will be display
D. Click the Report Button
E. A particular Report is display



Patient ID	Patient Name	New IC NO	Old IC NO	Date of Birth	Sex	Ethnic	Site of Myocardial Infarc
36	Mahammad Khir Bin Md Zir	700328025687	A1631150	28/03/1970	M	MAL	INF
10	Mahmud Bin Abdul Karim	500525025149		25/05/1950	M	MAL	ANT
34	Mohamad Salleh Bin Hamic	451215025217	3541230	15/12/1945	M	MAL	ANT
21	Mohammad Arshad Bin Haj	191213025039	0806603	13/12/1919	M	MAL	INF
33	Mohd Khairuddin Bin Abdu	710711025201	a1878245	11/07/1971	M	MAL	INF
1	Mohd Said Bin Abu Bakar	500911025171	4044271	11/09/1950	M	MAL	ANT
16	Mohd Zubir Bin Abd Hamic	590502025009	5672854	02/05/1959	M	MAL	ANT

11.2 Report on Patient Detail Information



ALOR SETAR HOSPITAL
YOUNG MYOCARDIAL INFARCTION INFOMATION SYSTEM
PATIENT DETAIL INFORMATION (REGISTRY) REPORT

Date 11/09/2000

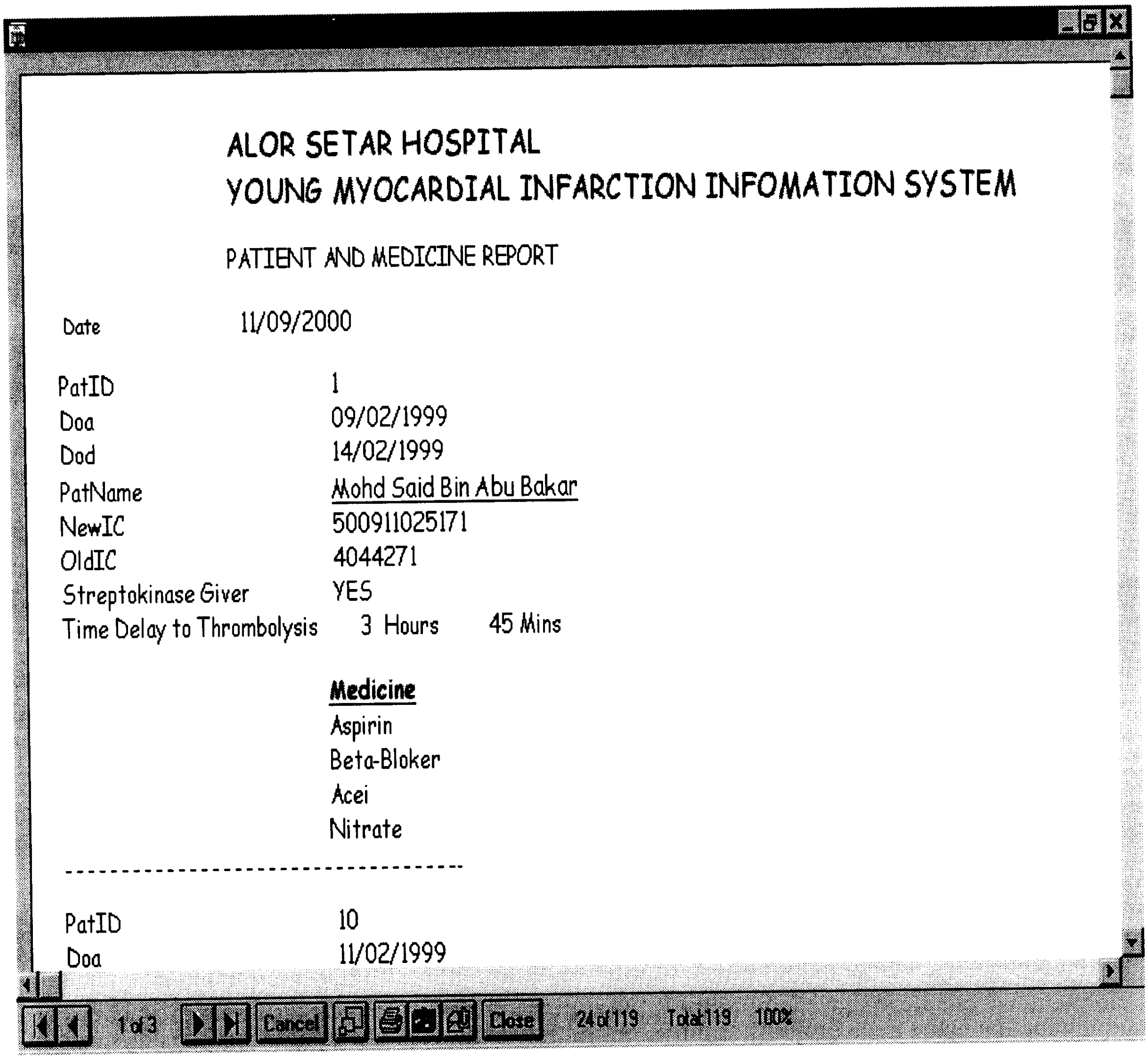
PATIENT BASIC INFOMATION

Patient ID	1
Doa	09/02/1999
Dod	14/02/1999
Patient Name	Mohd Said Bin Abu Bakar
NewIC	500911025171
OldIC	4044271
Dob	11/09/1950
Sex	Male
Ethnic	Malay

SITE OF MYOCARDIAL INFARCTION
Anterior/Anteroseptal/Anterolateral

RISK FACTOR		INVESTIGATION (ON ADMISSION)	
Smoker	YES	Total Colesterol	4.70
HyperTension	YES	Triglycerides	2.90
Diabetes Mellitus	NO	Blood Glucose	7.40

11.3 Report on Patient and Medical



ALOR SETAR HOSPITAL

YOUNG MYOCARDIAL INFARCTION INFOMATION SYSTEM

PATIENT AND MEDICINE REPORT

Date	11/09/2000
PatID	1
Doa	09/02/1999
Dod	14/02/1999
PatName	Mohd Said Bin Abu Bakar
NewIC	500911025171
OldIC	4044271
Streptokinase Giver	YES
Time Delay to Thrombolysis	3 Hours 45 Mins

Medicine

Aspirin

Beta-Bloker

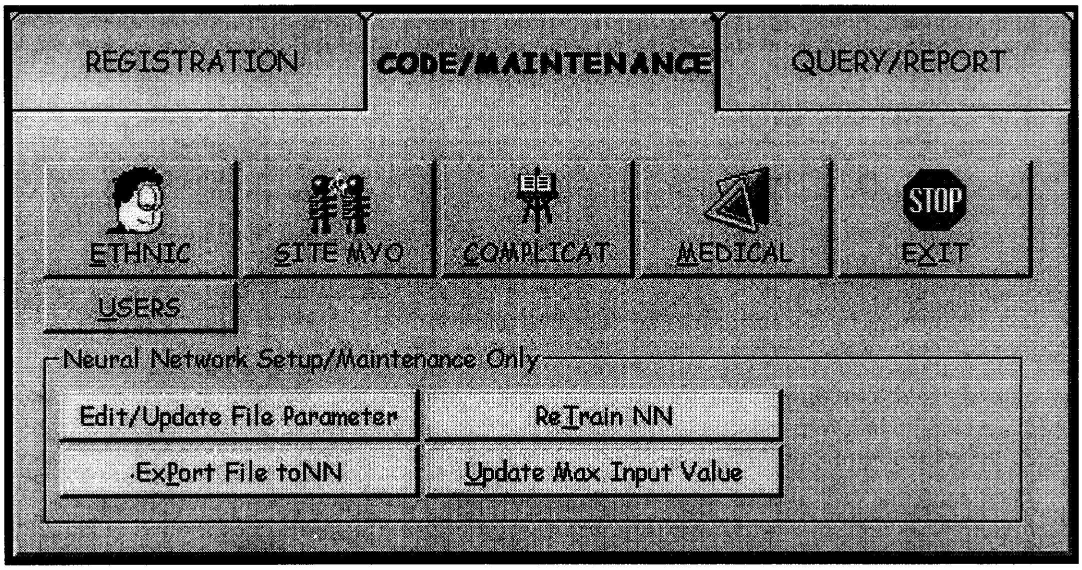
Acei

Nitrate

PatID	10
Doa	11/02/1999

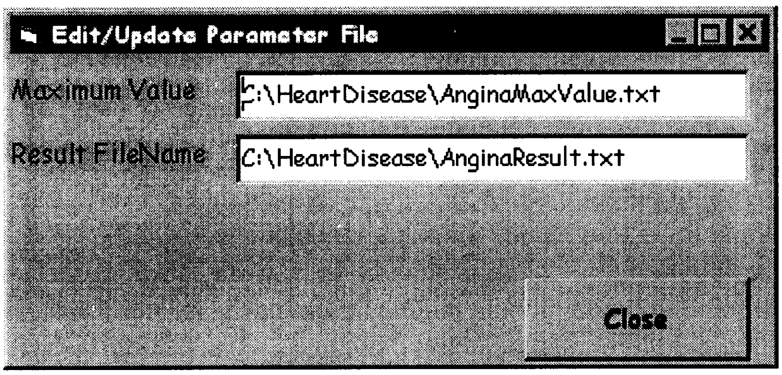
12.0 Neural Network Simulator

- Edit/Update File Parameter
- Export File to NN Format
- Update Max Input Value
- Retrain NN



12.1 Edit/Update Parameter File

- Use to link the program to the Maximum Value File name in the Prediction System (PS)
- Use to link the program to the NN Weight File name, where the actual weight will be installed by Neural Network after the training and will be called by Prediction System (PS)
- Edit the filename is necessary (Suggested to keep the existing one)

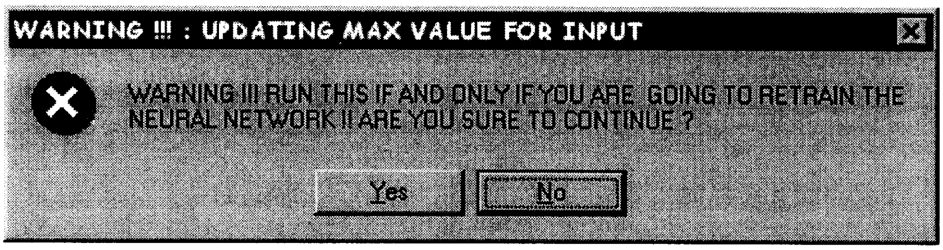


12.2 Export File to NN format

- Must be done before Retrain NN
- Click the Export File button

12.3 Update Maximum Value

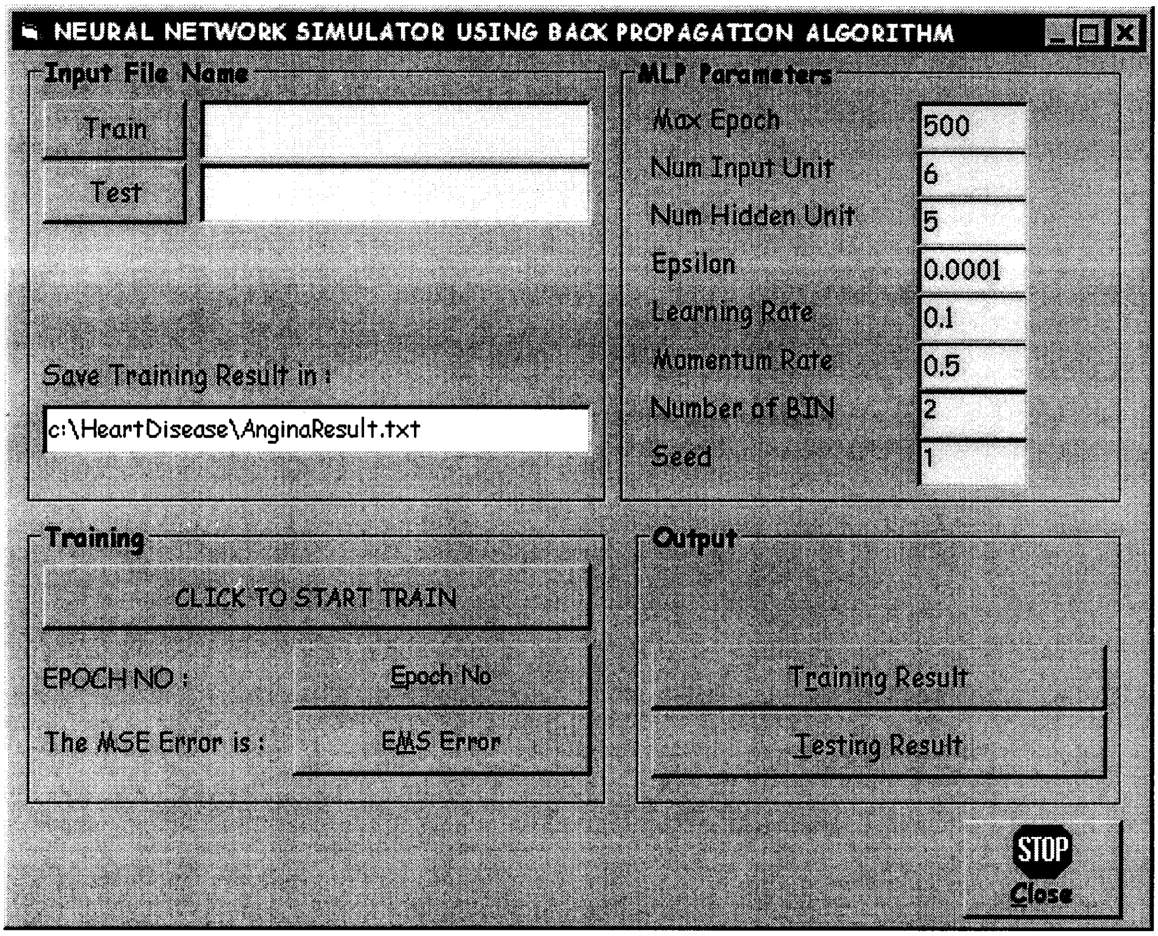
- To update the Maximum Value for a particular variable
- Can be done before or after Retrain NN
- Click the Update Max Value button



12.4 Retrain NN

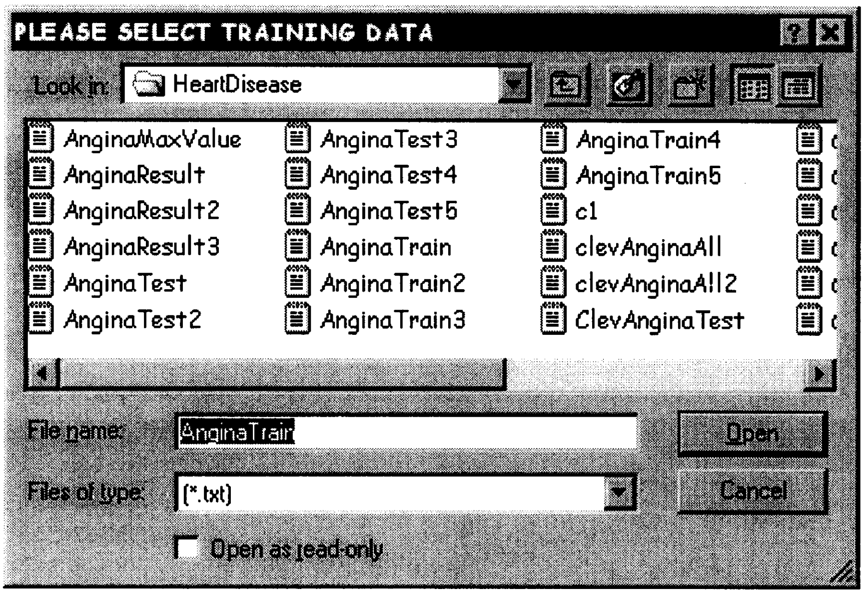
Basically NN simulator consist of 4 major part as follows:

- Input File Name
- MLP Parameters
- Training Session
- Display Output

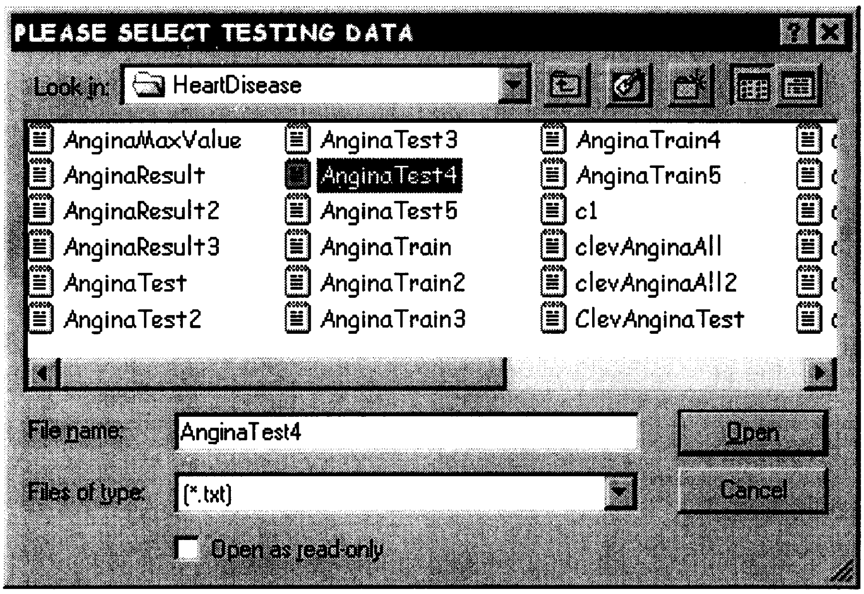


12.4.1 Input File Name

A. Enter Train button to set Training File Name
B. Select file from the Dialog Box
C. Click Open to confirm Training File



D. Enter Test button to set Testing File Name
E. Select file from the Dialog Box
F. Click Open to confirm Testing File



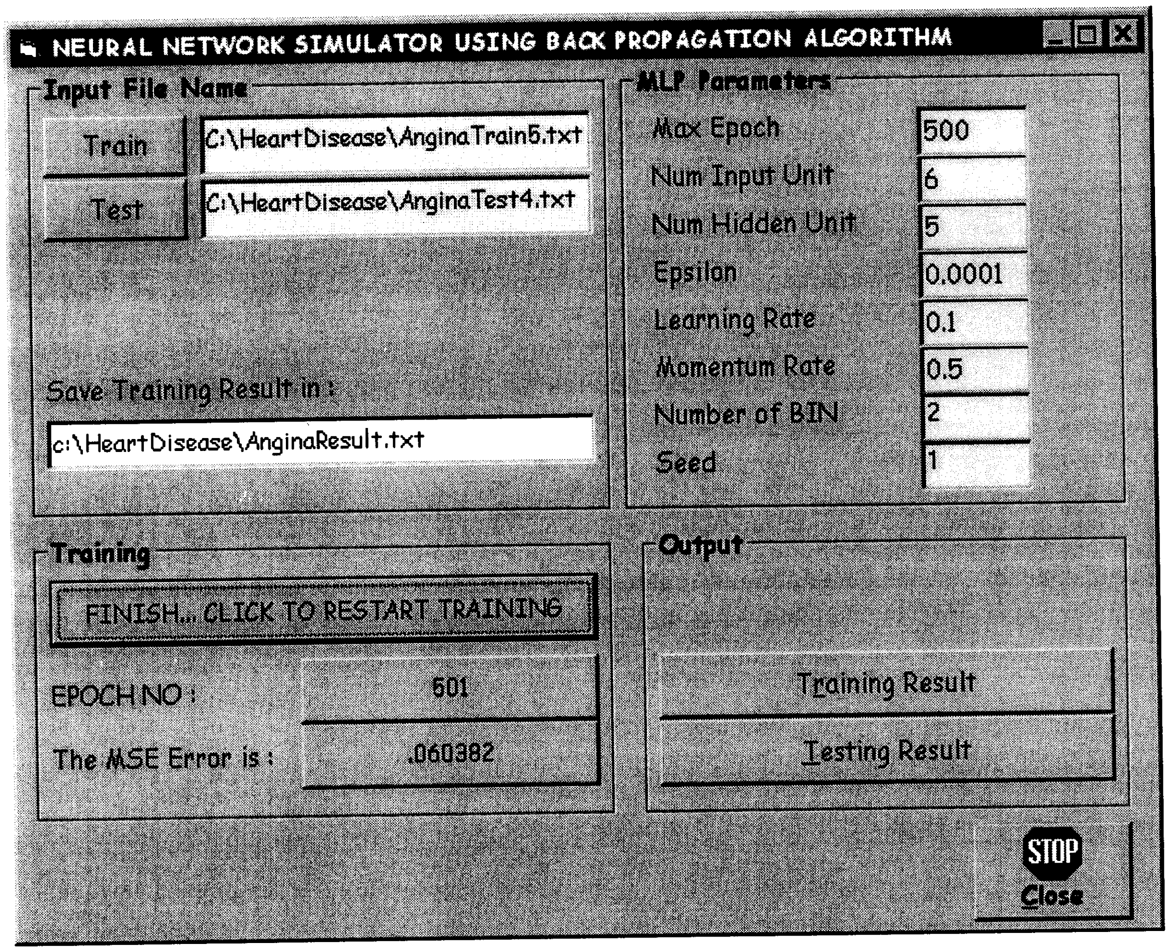
G. Enter Filename for Result (Weight File). The Filename must be the same as the one in the Input Parameter File Screen (Suggested not to make any changes)

12.4.2 MLP Parameter

A. Enter Epoch (Number of Iteration, default is 500)
B. Enter Input Unit (Default is 6)
C. Enter Hidden Unit (Default is 5)
D. Enter Epsilon (Error Tolerance, default is 0.0001)
E. Enter Learning rate (Default is 0.1)
F. Enter Momentum rate (Default is 0.5)
G. Enter Number of Bin (Default is 2)
H. Enter Seed number (Default is 1)

The default value is the current best model for the study

User can try different set of values, depending on their suitability.



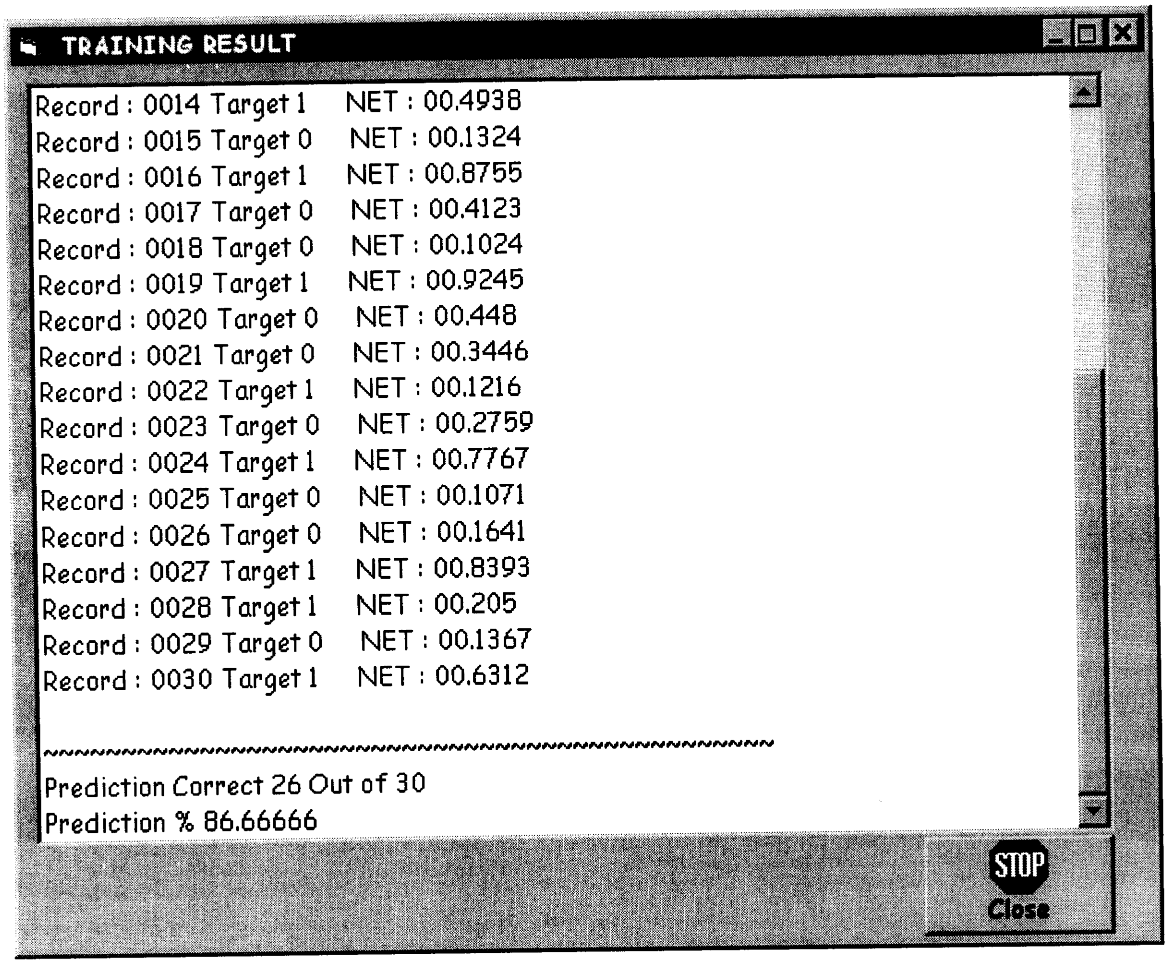
12.4.3 Training Session

- Once the input filename and the MLP parameter is completed, then the network can be train by clicking the Training button
- During the training, user can monitor the Epoch Number and the MSE
- Once the training is complete, the button caption will indicate to 'FINISH …' and user can proceed to see the Result/Output

12.4.4 Output

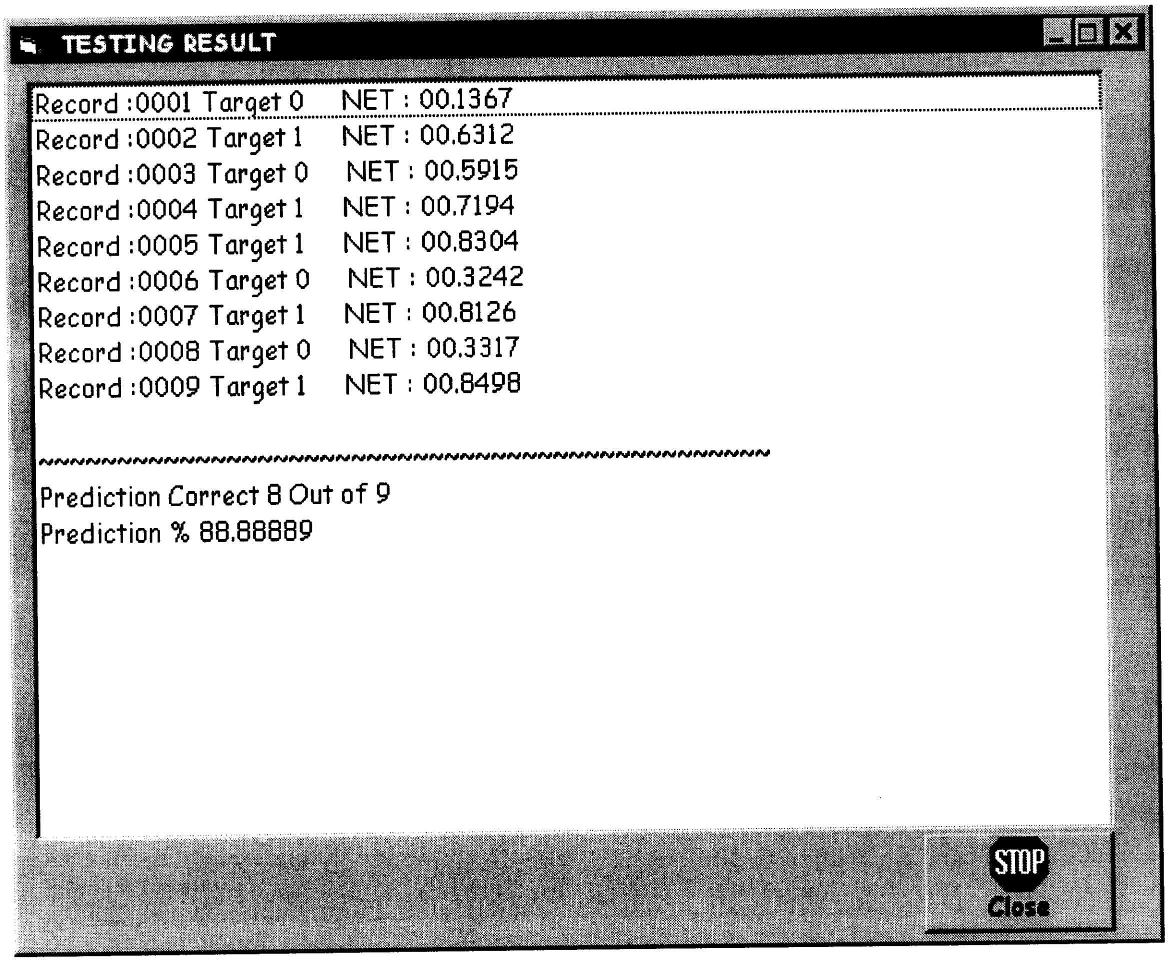
Training Result

A. Click the Training Result button to see the output for training patterns
B. Training Result screen is display
C. User can browse through the data
   - The prediction result is stated at the end of list



Testing result

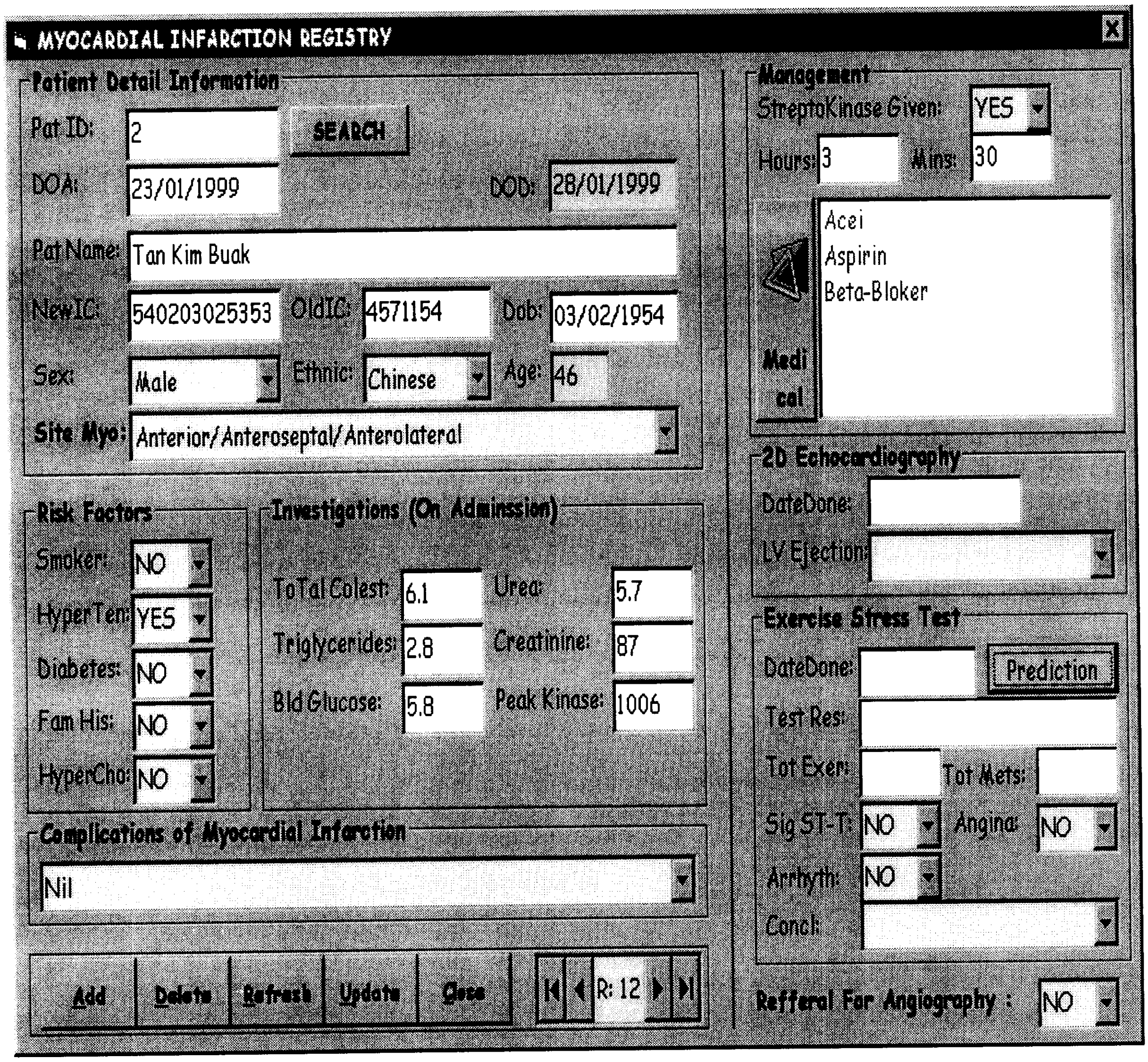
A. Click the Training Result button to see the output for training patterns
B. Training Result screen is display
C. User can browse through the data
   - The prediction result is stated at the end of list



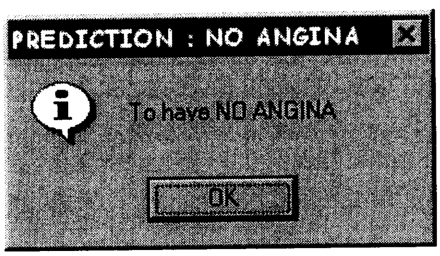
13.0 Prediction

A. Enter patient data as descried in ADD new patient in section 9.1
B. Click on Prediction Button
C. Prediction Result is display

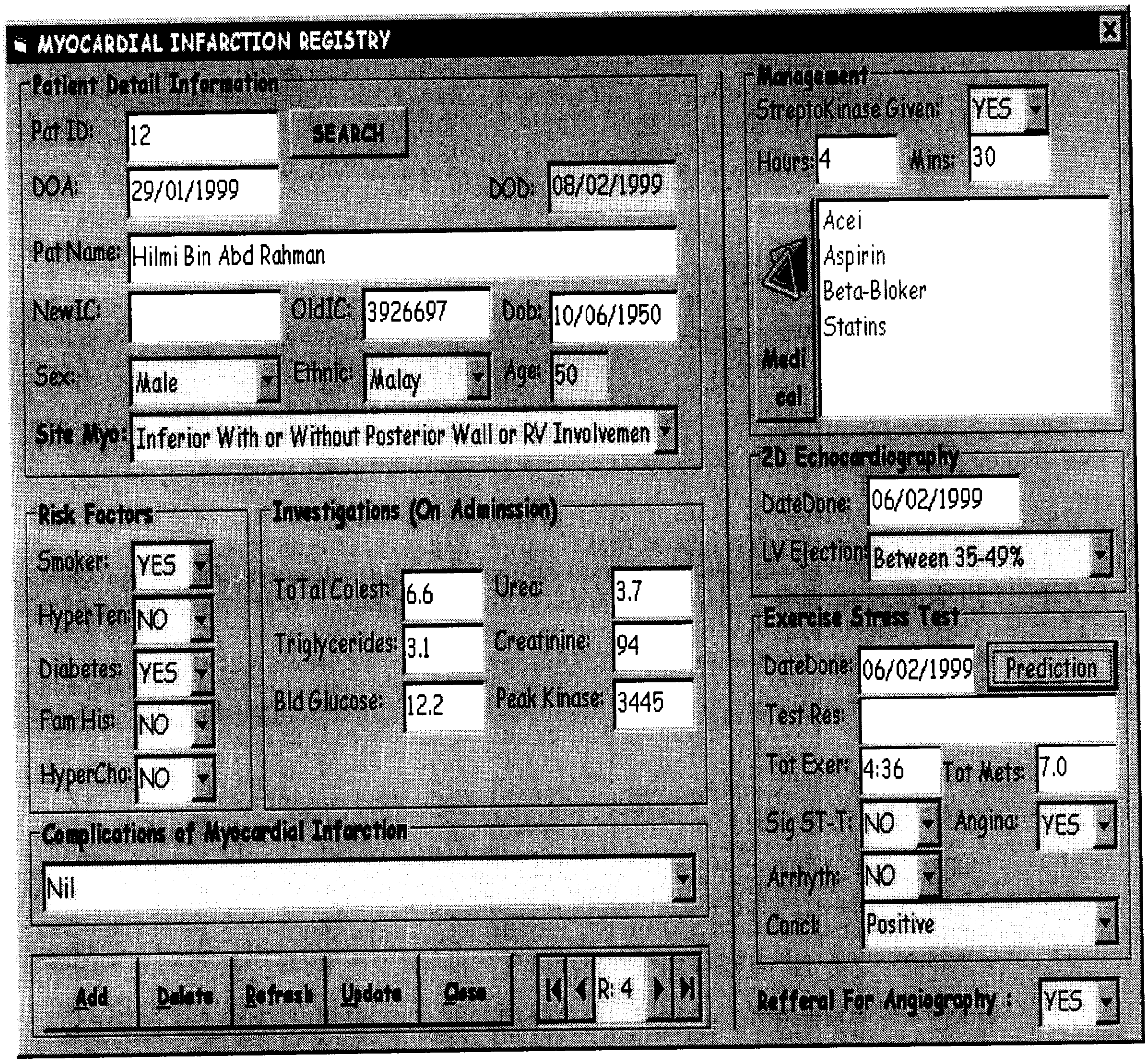
13.1 Example of Prediction on Patient Without Angina



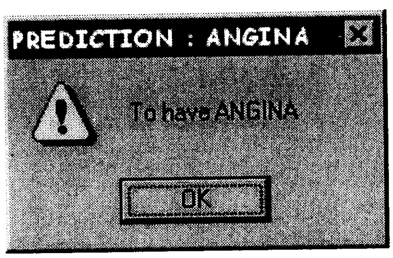
- Once the Prediction button is click, the next Dialog Box is Display
- The patient is predicted to have NO angina



13.2 Example of Prediction on Patient With Angina



- Once the Prediction button is click, the next Dialog Box is Display
- The patient is predicted to have angina



Appendix G Programming Manual

FRMCOMP.FRM

```
Private Sub cmdAdd_Click()
  On Local Error Resume Next
  datPrimaryRS.Recordset.AddNew
End Sub

Private Sub cmdDelete_Click()
  On Local Error Resume Next
  msg = "Are you to DELETE the Record ?"   ' Define message.
  Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
  Title = "WARNING : RECORD DELETION"  ' Define title.
  Response = MsgBox(msg, Style, Title)
  If Response = vbYes Then   ' User chose Yes.
    With datPrimaryRS.Recordset
     .Delete
     .MoveNext
     If .EOF Then .MoveLast
    End With
  End If
End Sub

Private Sub cmdRefresh_Click()
  On Local Error Resume Next
  'This is only needed for multi user apps
  datPrimaryRS.Refresh
End Sub

Private Sub cmdUpdate_Click()
  On Local Error Resume Next
  datPrimaryRS.UpdateRecord
  datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified
  DBGrid1.ReBind
End Sub

Private Sub cmdClose_Click()
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
  Unload Me
End Sub

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
    On Local Error Resume Next
  'This is where you would put error handling code
  'If you want to ignore errors, comment out the next line
  'If you want to trap them, add code here to handle them
  MsgBox "Data error event hit err:" & Error$(DataErr)
  Response = 0  'Throw away the error
End Sub
```

```
Private Sub datPrimaryRS_Reposition()
 On Local Error Resume Next
 Screen.MousePointer = vbDefault
 On Error Resume Next
 'This will display the current record position for dynasets and snapshots
 datPrimaryRS.Caption = "R: " & (datPrimaryRS.Recordset.AbsolutePosition + 1)
End Sub

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
 On Local Error Resume Next
 If Save = True Then
 Style = vbYesNo + vbCritical + vbDefaultButton2
   ir = MsgBox("Data Have Been CHANGED, Are You SURE to SAVE CHANGES?", Style, "DATA
CHANGED " & Me.Caption)
   If ir = vbNo Then
     Save = False
     datPrimaryRS.UpdateControls
   End If
 End If
End Sub

Private Sub Form_Load()
   On Local Error Resume Next
   datPrimaryRS.DatabaseName = gstrDBName
End Sub

Private Sub Form_Unload(Cancel As Integer)
 On Local Error Resume Next
 Screen.MousePointer = vbDefault
End Sub
```

FRMETHNIC.FRM

```
Private Sub cmdAdd_Click()
 On Local Error Resume Next
 datPrimaryRS.Recordset.AddNew
End Sub

Private Sub cmdDelete_Click()
  On Local Error Resume Next
 msg = "Are you to DELETE the Record ?"   ' Define message.
 Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
 Title = "WARNING : RECORD DELETION"  ' Define title.
 Response = MsgBox(msg, Style, Title)
 If Response = vbYes Then    ' User chose Yes.
   With datPrimaryRS.Recordset
    .Delete
    .MoveNext
    If .EOF Then .MoveLast
   End With
 End If
End Sub
```

```
Private Sub cmdRefresh_Click()
  On Local Error Resume Next
  'This is only needed for multi user apps
  datPrimaryRS.Refresh
End Sub

Private Sub cmdUpdate_Click()
  On Local Error Resume Next
  datPrimaryRS.UpdateRecord
  datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified
  DBGrid1.ReBind
End Sub

Private Sub cmdClose_Click()
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
  Unload Me
End Sub

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
    On Local Error Resume Next
  'This is where you would put error handling code
  'If you want to ignore errors, comment out the next line
  'If you want to trap them, add code here to handle them
  MsgBox "Data error event hit err:" & Error$(DataErr)
  Response = 0  'Throw away the error
End Sub

Private Sub datPrimaryRS_Reposition()
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
  On Error Resume Next
  'This will display the current record position for dynasets and snapshots
  datPrimaryRS.Caption = "R: " & (datPrimaryRS.Recordset.AbsolutePosition + 1)
End Sub

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
  On Local Error Resume Next
  If Save = True Then
  Style = vbYesNo + vbCritical + vbDefaultButton2
  ir = MsgBox("Data Have Been CHANGED, Are You SURE to SAVE CHANGES?", Style, "DATA
CHANGED " & Me.Caption)
     If ir = vbNo Then
       Save = False
       datPrimaryRS.UpdateControls
     End If
   End If
End Sub

Private Sub Form_Load()
    On Local Error Resume Next
    datPrimaryRS.DatabaseName = gstrDBName
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
 On Local Error Resume Next
 Screen.MousePointer = vbDefault
End Sub
```

FRMFLASH.FRM

```
Private Sub Image1_Click()
 On Local Error Resume Next
 Unload Me
 frmPengguna.Show
End Sub

Private Sub Timer1_Timer()
 On Local Error Resume Next
 Unload Me
 frmPengguna.Show
End Sub
```

FRMKOMED.FRM

```
Private Sub Command1_Click()
  On Local Error Resume Next
  'Clear table PATIENTMANAG
  PatientManag.RecordSource = "Select * from PatientManag WHERE PatID = '" &
frmPatient.txtFields(0) & "'"
  PatientManag.Refresh
  PatientManag.Recordset.MoveFirst
  Do Until PatientManag.Recordset.EOF    ' Do until end of file.
     PatientManag.Recordset.Delete
     PatientManag.Recordset.MoveNext
  Loop
  For i = 0 To DBGrid1.SelBookmarks.Count - 1
    X2 = DBGrid1.Columns(0).CellValue(DBGrid1.SelBookmarks(i))
    'update table PATIENTMANAG
    PatientManag.Recordset.AddNew
    PatientManag.Recordset(0) = frmPatient.txtFields(0)
    PatientManag.Recordset(1) = X2
    PatientManag.Recordset.Update
   Next i
   Unload Me
End Sub


Private Sub Command2_Click()
   On Local Error Resume Next
   frmMed.Show 1
   Data1.Refresh
   DBGrid1.ReBind
End Sub

Private Sub Command3_Click()
   On Local Error Resume Next
   Unload Me
End Sub
```

```
Private Sub DBGrid1_DblClick()
On Local Error Resume Next
  'Clear table PATIENTMANAG
  PatientManag.RecordSource = "Select * from PatientManag WHERE PatID = '" &
frmPatient.txtFields(0) & "'"
  PatientManag.Refresh
  PatientManag.Recordset.MoveFirst
  Do Until PatientManag.Recordset.EOF    ' Do until end of file.
     PatientManag.Recordset.Delete
     PatientManag.Recordset.MoveNext
  Loop
    For i = 0 To DBGrid1.SelBookmarks.Count - 1
     X2 = DBGrid1.Columns(0).CellValue(DBGrid1.SelBookmarks(i))
     'update table PATIENTMANAG
     PatientManag.Recordset.AddNew
     PatientManag.Recordset(0) = frmPatient.txtFields(0)
     PatientManag.Recordset(1) = X2
     PatientManag.Recordset.Update
    Next i
    Unload Me
End Sub


Private Sub Form_Load()
On Local Error Resume Next
   Data1.DatabaseName = gstrDBName
   PatientManag.DatabaseName = gstrDBName
End Sub

Private Sub Text2_Change()
On Local Error Resume Next
diga = Text2 & "*"
Data1.RecordSource = "Select * from Managment WHERE ManagName like '" & diga & "' order by
ManagName"
Data1.Refresh
End Sub
```

FRMMAIN.FRM

```
Private Sub Command1_Click()
 On Local Error Resume Next
 frmPatient.Show
End Sub

Private Sub Command10_Click()
 On Local Error Resume Next
 End
End Sub

Private Sub Command12_Click()
  On Local Error Resume Next
  reportno = 2
  frmQuery.Caption = Command12.ToolTipText
  frmQuery.Command2.ToolTipText = Command12.ToolTipText
  frmQuery.Show 1
End Sub
```

```
Private Sub Command11_Click()
  On Local Error Resume Next
    End
End Sub

Private Sub Command13_Click()
   On Local Error Resume Next
   reportno = 1
   frmQuery.Caption = Command13.ToolTipText
   frmQuery.Command2.ToolTipText = Command13.ToolTipText
   frmQuery.Show 1
End Sub

Private Sub Command14_Click()
  frmMLP002a.Show
End Sub

Private Sub Command15_Click()
   frmParameter.Show
End Sub

Private Sub Command2_Click()
  On Local Error Resume Next
  frmMed.Show
End Sub

Private Sub Command3_Click()
  On Local Error Resume Next
  frmPengguna1.Show
End Sub

Private Sub Command4_Click()
  On Local Error Resume Next
  frmSite.Show
End Sub

Private Sub Command5_Click()
  On Local Error Resume Next
  frmEthnic.Show
End Sub

Private Sub Command6_Click()
  On Local Error Resume Next
  frmComp.Show
End Sub

Private Sub Command7_Click()
   On Local Error Resume Next
   msg1 = "WARNING !!! RUN THIS IF AND ONLY IF YOU ARE "
   msg2 = msg1 & " GOING TO RETRAIN THE NEURAL NETWORK !!"
   msg = msg2 & " ARE YOU SURE TO CONTINUE ? "
   Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
   Title = "WARNING !!! : UPDATING MAX VALUE FOR INPUT"  ' Define title.
   Response = MsgBox(msg, Style, Title)
   If Response = vbNo Then    ' User chose Yes.
     Exit Sub
```

```
End If
Open NNMaxValueFile For Output As #1
MaxChol = 0
MaxAge = 0
TotChol = 0
TotAge = 0
TotRecord = 0
Data1.Recordset.MoveFirst
Do Until Data1.Recordset.EOF   ' Do until end of file.
    If Data1.Recordset("InvColest") >= MaxChol Then MaxChol = Data1.Recordset("InvColest")
    If Data1.Recordset("Age") >= MaxAge Then MaxAge = Data1.Recordset("Age")
    TotChol = TotChol + Data1.Recordset("InvColest")
    TotAge = TotAge + Data1.Recordset("Age")
    TotRecord = TotRecord + 1
    Data1.Recordset.MoveNext
Loop
MeanChol = (TotChol / TotRecord) / MaxChol
MeanAge = (TotAge / TotRecord) / MaxAge
Write #1, "MaxChol ", MaxChol
Write #1, "MaxAge ", MaxAge
Write #1, "MeanChol ", MeanChol
Write #1, "MeanAge ", MeanAge
Close #1
End Sub

Private Sub Command8_Click()
  Command8.Caption = "Processing..."
  'selecting the database location
  Open "c:\heartdisease\heart.txt" For Output As #1
  ' Find the MaxValue
  MaxChol = 0
  MaxAge = 0
  TotChol = 0
  TotAge = 0
  TotRecord = 0
  Data1.Recordset.MoveFirst
  Do Until Data1.Recordset.EOF   ' Do until end of file.
     If Data1.Recordset("InvColest") >= MaxChol Then MaxChol = Data1.Recordset("InvColest")
     If Data1.Recordset("Age") >= MaxAge Then MaxAge = Data1.Recordset("Age")
     TotChol = TotChol + Data1.Recordset("InvColest")
     TotAge = TotAge + Data1.Recordset("Age")
     TotRecord = TotRecord + 1
     Data1.Recordset.MoveNext
  Loop
  MeanChol = (TotChol / TotRecord) / MaxChol
  MeanAge = (TotAge / TotRecord) / MaxAge
  ' Write Normalized Data
  Data1.Recordset.MoveFirst
  Do Until Data1.Recordset.EOF   ' Do until end of file.
     i1 = Data1.Recordset("RiskSmoker")
     i2 = Data1.Recordset("RiskHyperTens")
     i3 = Data1.Recordset("RiskDiabet")
     i4 = Data1.Recordset("RiskFamHis")
     i5 = Data1.Recordset("RiskHyperChole")
     i6 = Data1.Recordset("InvColest")
     i7 = Data1.Recordset("MagStreKin")
```

```
    i8 = Data1.Recordset("Age")
    i9 = Data1.Recordset("RiskStAngina")
    If i1 = "YES" Then i1 = 1 Else i1 = 0
    If i2 = "YES" Then i2 = 1 Else i2 = 0
    If i3 = "YES" Then i3 = 1 Else i3 = 0
    If i4 = "YES" Then i4 = 1 Else i4 = 0
    If i5 = "YES" Then i5 = 1 Else i5 = 0
    If i6 = 0 Then i6 = MeanChol Else i6 = i6 / MaxChol
    If i7 = "YES" Then i7 = 1 Else i7 = 0
    If i8 = 0 Then i8 = MeanAge Else i8 = i8 / MaxAge
    If i9 = "YES" Then i9 = 1 Else i9 = 0
    Write #1, i1, i2, i3, i4, i5, i6, i7, i8, i9
    Data1.Recordset.MoveNext
  Loop
  Close #1
  Command8.Caption = "Ex&Port File toNN"
End Sub

Private Sub Command9_Click()
 On Local Error Resume Next
 End
End Sub

Private Sub Form_Load()
   On Local Error Resume Next
   Data1.DatabaseName = gstrDBName
   If penggunax = "admin" Then
      Command3.Visible = True
      Frame1.Visible = True
   End If
   ' Initialize file name for NN
   NNMaxValueFile = "C:\HeartDisease\AnginaMaxValue.txt"
   NNResultFile = "C:\HeartDisease\AnginaResult.txt"
End Sub

Private Sub Timer1_Timer()
On Local Error Resume Next
   dayx = Format(Date, "dddd")
   Datex = Format(Date, "d mmm yyyy")
   Label1 = dayx & " : " & Datex & " : " & Time
End Sub
```

FRMMED.FRM

```
Private Sub cmdAdd_Click()
 On Local Error Resume Next
 datPrimaryRS.Recordset.AddNew
End Sub

Private Sub cmdDelete_Click()
  On Local Error Resume Next
 msg = "Are you to DELETE the Record ?"  ' Define message.
 Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
 Title = "WARNING : RECORD DELETION"  ' Define title.
 Response = MsgBox(msg, Style, Title)
```

```
 If Response = vbYes Then    ' User chose Yes.
  With datPrimaryRS.Recordset
   .Delete
   .MoveNext
   If .EOF Then .MoveLast
  End With
 End If
End Sub

Private Sub cmdRefresh_Click()
 On Local Error Resume Next
 'This is only needed for multi user apps
 datPrimaryRS.Refresh
End Sub

Private Sub cmdUpdate_Click()
 On Local Error Resume Next
 datPrimaryRS.UpdateRecord
 datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified
 DBGrid1.ReBind
End Sub

Private Sub cmdClose_Click()
 On Local Error Resume Next
 Screen.MousePointer = vbDefault
 Unload Me
End Sub

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
   On Local Error Resume Next
 'This is where you would put error handling code
 'If you want to ignore errors, comment out the next line
 'If you want to trap them, add code here to handle them
 MsgBox "Data error event hit err:" & Error$(DataErr)
 Response = 0  'Throw away the error
End Sub

Private Sub datPrimaryRS_Reposition()
 On Local Error Resume Next
 Screen.MousePointer = vbDefault
 On Error Resume Next
 'This will display the current record position for dynasets and snapshots
 datPrimaryRS.Caption = "R: " & (datPrimaryRS.Recordset.AbsolutePosition + 1)
End Sub

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
   On Local Error Resume Next
 If Save = True Then
 Style = vbYesNo + vbCritical + vbDefaultButton2
   ir = MsgBox("Data Have Been CHANGED, Are You SURE to SAVE CHANGES?", Style, "DATA
CHANGED " & Me.Caption)
   If ir = vbNo Then
     Save = False
     datPrimaryRS.UpdateControls
   End If
 End If
```

```
End Sub

Private Sub Form_Load()
   On Local Error Resume Next
   datPrimaryRS.DatabaseName = gstrDBName
End Sub

Private Sub Form_Unload(Cancel As Integer)
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
End Sub
```

FRMMLP002a.FRM

```
Private Sub Command1_Click()
   Command1.Caption = "TRAINING IN PROGRESS ... PLEASE WAIT ."
   SetParameter
   ResultFileName = Text7.Text
   Open ResultFileName For Output As #2
   InitWeight
   training
   WriteModelWeight
   Close #2
   Beep
   Command1.Caption = "FINISH... CLICK TO RESTART TRAINING"
End Sub

Private Sub Command4_Click()
   'call function open train file
   OpenTrainFile
End Sub

Private Sub Command5_Click()
    Unload Me
End Sub

Private Sub Command6_Click()
   'call function open Test file
   OpenTestFile
End Sub

Private Sub Command7_Click()
   TrainingResult
   frmMLP002b.Show
   frmMLP002b.Caption = " TRAINING RESULT "
   frmMLP002b.List1.AddItem " "
   frmMLP002b.List1.AddItem "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
   frmMLP002b.List1.AddItem "Prediction Correct " & PercentCorrect & " Out of " & NumOfTrain
   frmMLP002b.List1.AddItem "Prediction % " & (PercentCorrect / NumOfTrain) * 100
End Sub

Private Sub Command8_Click()
   TestingResult
   frmMLP002b.Show
   frmMLP002b.Caption = " TESTING RESULT "
   frmMLP002b.List1.AddItem " "
```

```
   frmMLP002b.List1.AddItem "~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~"
   frmMLP002b.List1.AddItem "Prediction Correct " & PercentCorrect & " Out of " & NumOfTest
   frmMLP002b.List1.AddItem "Prediction % " & (PercentCorrect / NumOfTest) * 100
End Sub

Private Sub Text10_Change()
   TestFileName = Text10
End Sub

Private Sub Text8_Change()
   TrainFileName = Text8
End Sub

Private Sub Text9_Change()
   ValidationFileName = Text9
End Sub
```

FRMMLP002b.FRM

```
Private Sub Command1_Click()
   Unload Me
End Sub
```

FRMPARAMETER.FRM

```
Private Sub Command1_Click()
   Unload Me
End Sub

Private Sub Form_Load()
   Text1.Text = NNMaxValueFile
   Text3.Text = NNResultFile
End Sub

Private Sub Text1_Change()
   NNMaxValueFile = Text1
End Sub

Private Sub Text3_Change()
   NNResultFile = Text3
End Sub
```

FRMPATIENT.FRM

```
Private Sub cmdAdd_Click()
   On Local Error Resume Next
  newrekod = "T"
  datPrimaryRS.Recordset.AddNew
  Combo1 = "NO"
  Combo2 = "NO"
  Combo3 = "NO"
  Combo4 = "NO"
  Combo5 = "NO"
  Combo6 = "NO"
  Combo7 = "NO"
  Combo8 = "NO"
```

```
Combo9 = "NO"
Combo10 = "NO"
DBCombo5 = "NOT DONE"
 Data1.Recordset.MoveFirst
 big = Int(Data1.Recordset(0))
 Do Until Data1.Recordset.EOF    ' Do until end of file.
     Data1.Recordset.MoveNext ' Move to next record.
     newdata = Int(Data1.Recordset(0))
     If newdata >= big Then big = newdata
 Loop
 txtFields(0) = big + 1
End Sub

Private Sub cmdDelete_Click()
 On Local Error Resume Next
 newrekod = "F"
 msg = "Are you Sure to DELETE the Record ?"   ' Define message.
 Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
 Title = "WARNING : RECORD DELETION"  ' Define title.
 Response = MsgBox(msg, Style, Title)
 If Response = vbYes Then    ' User chose Yes.
   With datPrimaryRS.Recordset
   .Delete
   .MoveNext
   If .EOF Then .MoveLast
   End With
  End If
  datPrimaryRS.Refresh
 End Sub

Private Sub cmdRefresh_Click()
 On Local Error Resume Next
newrekod = "F"
 'This is only needed for multi user apps
 datPrimaryRS.Refresh
End Sub

Private Sub cmdUpdate_Click()
 On Local Error Resume Next
 newrekod = "F"
 datPrimaryRS.UpdateRecord
 datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified
 Data1.Refresh
End Sub

Private Sub cmdClose_Click()
 On Local Error Resume Next
 Screen.MousePointer = vbDefault
 Unload Me
End Sub

Private Sub Combo12_Click()
 On Local Error Resume Next
 If str = 0 Then strx = "PatID = '" & Combo12 & "'"
 If str = 1 Then strx = "PatName = '" & Combo12 & "'"
 If str = 2 Then strx = "NewIC = '" & Combo12 & "'"
```

```
If str = 3 Then strx = "OldIC = '" & Combo12 & "'"
datPrimaryRS.Recordset.FindFirst strx
 If datPrimaryRS.Recordset.NoMatch Then
   MsgBox "No records found with " & _
   strx & "."
   Exit Sub
 End If
 Combo12.Visible = False
End Sub

Private Sub Combo13_Click()
  On Local Error Resume Next
  Combo13.Visible = False
  Combo12.Visible = True
  Combo12.Clear
  Data1.RecordSource = "select * from Patient"
  Data1.Refresh
  Do Until Data1.Recordset.EOF    ' Do until end of file.
    If Combo13 = "Patient No" Then
      Combo12.AddItem Data1.Recordset(0)
      str = 0
      Combo12.ToolTipText = "CLICK TO SELECT PATIENT NO OR ENTER A SPECIFIC
PATIENT NO"
    End If
    If Combo13 = "Patient Name" Then
      Combo12.AddItem Data1.Recordset(1)
      str = 1
      Combo12.ToolTipText = "CLICK TO SELECT PATIENT NAME OR ENTER A SPECIFIC
PATIENT NAME"
    End If
    If Combo13 = "Patient NewIC" Then
      Combo12.AddItem Data1.Recordset(2)
      str = 2
      Combo12.ToolTipText = "CLICK TO SELECT PATIENT NEWIC OR ENTER A SPECIFIC
PATIENT NEWIC"
    End If
    If Combo13 = "Patient OldIC" Then
      Combo12.AddItem Data1.Recordset(3)
      str = 3
      Combo12.ToolTipText = "CLICK TO SELECT PATIENT OLDIC OR ENTER A SPECIFIC
PATIENT OLDIC"
    End If
    Data1.Recordset.MoveNext    ' Move to next record.
  Loop
  Combo12.SetFocus
End Sub

Private Sub Command1_Click()
On Local Error Resume Next
  If newrekod = "T" Then
    'MsgBox "New RECORD"
    datPrimaryRS.UpdateRecord
    datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified
  End If
  frmKodMed.Show 1
End Sub
```

```
Private Sub Command2_Click()
   On Local Error Resume Next
   Combo13.Visible = True
   Combo13.Text = "PICK CRITERIA"
End Sub

Private Sub Command3_Click()
   ' The Prediction Process
   ' Open NNResultFile to read the Calculated Weights
   ' and assign it to VAR
   Open NNResultFile For Input As #1
   Input #1, x, jumin ' Number of input Node
   Input #1, mm, jumhid ' Number of Hidden Node
   Dim WtInputToHidden() As Double
   Dim WtHiddenToOutput() As Double
   ReDim WtHiddenToOutput(jumhid)
   ReDim WtInputToHidden(jumin, jumhid)
   ' Assign Weights to INPUT-HIDDEN
   For i = 0 To jumin - 1
      For j = 0 To jumhid - 1
         Input #1, r1, r2, r3, r4
         WtInputToHidden(i, j) = r4
      Next j
   Next i
   ' Assign Weights to HIDDEN-OUTPUT
   For i = 0 To jumhid - 1
      Input #1, r1, r2, r3
      WtHiddenToOutput(i) = r3
   Next i
   Close #1
   ' Read MaxValue, Meanvalue for Cholestrol and Age
   Open NNMaxValueFile For Input As #2
   Input #2, mx2, MaxChol
   Input #2, mx3, MaxAge
   Input #2, mx4, MeanChol
   Input #2, mx5, MeanAge
   Close #2
   Dim Zin, Yin As Double ' Total summation of weighted input signals
   ' Calculate the activation of all nodes for the training pattern
   ' presented to the function
   Dim ArrayOfInput() As Variant
   ReDim ArrayOfInput(jumin)
   Dim Z() As Double
   ReDim Z(jumhid)
   ' Pre-Prcessed Data before being forward as INPUT in NN
   If Combo1 = "YES" Then smoker = 1 Else smoker = 0
   If Combo2 = "YES" Then Hpt = 1 Else Hpt = 0
   If Combo3 = "YES" Then Dm = 1 Else Dm = 0
   If Combo4 = "YES" Then Fhx = 1 Else Fhx = 0
   If txtFields(5) = 0 Then
      chol = MeanChol
      MsgBox chol
   Else
      chol = txtFields(5) / MaxChol
   End If
```

```
If chol > 1 Then chol = 1
If txtFields(20) = 0 Then
   age = MeanAge
   MsgBox age
Else
   age = txtFields(20) / MaxAge
End If
If age > 1 Then age = 1
ArrayOfInput(0) = smoker
ArrayOfInput(1) = Hpt
ArrayOfInput(2) = Dm
ArrayOfInput(3) = Fhx
ArrayOfInput(4) = chol
ArrayOfInput(5) = age
For j = 0 To jumhid - 1
   Zin = 0
   For i = 0 To jumin - 1
      Zin = Zin + ArrayOfInput(i) * WtInputToHidden(i, j)
   Next i
   Z(j) = 1 / (1 + Exp(-Zin))
Next j
Yin = 0
For i = 0 To jumhid - 1
      Yin = Yin + Z(i) * WtHiddenToOutput(i)
Next i
Y = 1 / (1 + Exp(-Yin))
' POST PROCESSING - RESCALING THE OUTPUT
' If the CALCULATED OUTPUT is less then 0.5 then the
' patient will have NO ANGINA
' Else he/she will have ANGINA
If Y < 0.5 Then
      msg = "To have NO ANGINA"   ' Define message.
      Style = vbInformation  ' Define buttons.
      Title = "PREDICTION : NO ANGINA"  ' Define title.
      x = MsgBox(msg, Style, Title)
   Else
      msg = "To have ANGINA"   ' Define message.
      Style = vbExclamation   ' Define buttons.
      Title = "PREDICTION : ANGINA"  ' Define title.
      x = MsgBox(msg, Style, Title)
 End If
End Sub

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
  On Local Error Resume Next
 'This is where you would put error handling code
 'If you want to ignore errors, comment out the next line
 'If you want to trap them, add code here to handle them
 MsgBox "Data error event hit err:" & Error$(DataErr)
 Response = 0  'Throw away the error
End Sub

Private Sub datPrimaryRS_Reposition()
 On Local Error Resume Next
 Screen.MousePointer = vbDefault
 On Error Resume Next
```

```
  'This will display the current record position for dynasets and snapshots
  datPrimaryRS.Caption = "R: " & (datPrimaryRS.Recordset.AbsolutePosition + 1)
End Sub

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
   On Local Error Resume Next
   If Save = True Then
   Style = vbYesNo + vbCritical + vbDefaultButton2
    ir = MsgBox("Data Have Been CHANGED, Are You SURE to SAVE CHANGES?", Style, "DATA
CHANGED " & Me.Caption)
    If ir = vbNo Then
      Save = False
      datPrimaryRS.UpdateControls
    End If
  End If
End Sub

Private Sub DBCombo2_DblClick(Area As Integer)
      frmEthnic.Show 1
      Ethnic.Refresh
      DBCombo2.Refresh
End Sub

Private Sub DBCombo2_KeyPress(KeyAscii As Integer)
   On Local Error Resume Next
   If KeyAscii = 13 Then
     frmEthnic.Show 1
     Ethnic.Refresh
     DBCombo2.Refresh
   End If
End Sub

Private Sub DBCombo3_DblClick(Area As Integer)
   frmSite.Show 1
   SiteMyo.Refresh
   DBCombo3.Refresh
End Sub

Private Sub DBCombo3_KeyPress(KeyAscii As Integer)
  On Local Error Resume Next
  If KeyAscii = 13 Then
     frmSite.Show 1
     SiteMyo.Refresh
     DBCombo3.Refresh
   End If
End Sub

Private Sub DBCombo4_DblClick(Area As Integer)
      frmComp.Show 1
      Compication.Refresh
      DBCombo4.Refresh
End Sub
```

```
Private Sub DBCombo4_KeyPress(KeyAscii As Integer)
    On Local Error Resume Next
    If KeyAscii = 13 Then
     frmComp.Show 1
     Compication.Refresh
     DBCombo4.Refresh
   End If
End Sub

Private Sub Form_Activate()
      On Local Error Resume Next
      newrekod = "F"
      List1.Clear
      PatientManag.RecordSource = "Select * from PatientManagquery WHERE PatID = '" & txtFields(0)
& "'"
      PatientManag.Refresh
      Do Until PatientManag.Recordset.EOF   ' Do until end of file.
         List1.AddItem PatientManag.Recordset(4)
         PatientManag.Recordset.MoveNext
      Loop
End Sub

Private Sub Form_Load()
   On Local Error Resume Next
   RiskECG.DatabaseName = gstrDBName
   PatientManag.DatabaseName = gstrDBName
    Sex.DatabaseName = gstrDBName
    Ethnic.DatabaseName = gstrDBName
    SiteMyo.DatabaseName = gstrDBName
    Compication.DatabaseName = gstrDBName
    Data1.DatabaseName = gstrDBName
   datPrimaryRS.DatabaseName = gstrDBName
   Combo1.AddItem "YES"
   Combo1.AddItem "NO"
   Combo2.AddItem "YES"
   Combo2.AddItem "NO"
   Combo3.AddItem "YES"
   Combo3.AddItem "NO"
   Combo4.AddItem "YES"
   Combo4.AddItem "NO"
   Combo5.AddItem "YES"
   Combo5.AddItem "NO"
   Combo6.AddItem "YES"
   Combo6.AddItem "NO"
   Combo7.AddItem "YES"
   Combo7.AddItem "NO"
   Combo8.AddItem "YES"
   Combo8.AddItem "NO"
   Combo9.AddItem "YES"
   Combo9.AddItem "NO"
   Combo10.AddItem "YES"
   Combo10.AddItem "NO"
   Combo11.AddItem "Inclusive"
   Combo11.AddItem "Positive"
   Combo11.AddItem "Negative"
```

```
   Combo13.AddItem "Patient No"
   Combo13.AddItem "Patient Name"
   Combo13.AddItem "Patient NewIC"
   Combo13.AddItem "Patient OldIC"
End Sub

Private Sub Form_Unload(Cancel As Integer)
   On Local Error Resume Next
   Screen.MousePointer = vbDefault
End Sub

Private Sub txtFields_Change(Index As Integer)
   On Local Error Resume Next
   If Index = 0 Then
      List1.Clear
      PatientManag.RecordSource = "Select * from PatientManagquery WHERE PatID = '" & txtFields(0)
& "'"
      PatientManag.Refresh
      Do Until PatientManag.Recordset.EOF    ' Do until end of file.
         List1.AddItem PatientManag.Recordset(4)
      PatientManag.Recordset.MoveNext
      Loop
   End If
   If Index = 2 Then
      'assign sex from newic
      If Len(txtFields(2)) = 12 Then
      Sex = Right(txtFields(2), 1)
      If IsNumeric(Sex) Then
         If (Sex Mod 2) = 0 Then
            DBCombo1 = "Female"
         Else
            DBCombo1 = "Male"
         End If
         dob = Left(txtFields(2), 6)
         doby = Left(dob, 2)
         dobm = Left(dob, 4)
         dobm = Right(dobm, 2)
         dobd = Right(dob, 2)
         dob = dobd & "/" & dobm & "/" & doby
         dob = Format(dob, "dd/mm/yyyy")
         txtFields(4) = dob
      End If
      End If
   End If
   If Index = 4 Then
      date1 = txtFields(4)
      date2 = Now
      msg = DateDiff("yyyy", date1, date2)
      txtFields(20) = msg
   End If

End Sub
```

FRMPENGGUNA.FRM

```
Private Sub Command1_Click()
  On Local Error Resume Next
  Dim MyCriteria As String

 MyCriteria = "pengguna = '" & Text2 & "' and KataLaluan = '" & Text1 & "'" ' Set the criteria.
 Data1.Recordset.FindFirst MyCriteria   ' Find first matching record.
 If Not Data1.Recordset.NoMatch Then
  If LCase(Text2) = "admin" Then penggunax = "admin"
  Unload Me
  frmMain.Show
 Else
  msg = "Permission Denied , Try again ?"   ' Define message.
  Style = vbOKOnly + vbCritical  ' Define buttons.
  Title = "WARNING : UNAUTHORISED USER"  ' Define title.
  Response = MsgBox(msg, Style, Title)
 End If
 ' Check if record is found.
End Sub

Private Sub Command2_Click()
  On Local Error Resume Next
  Text1 = ""
  Text2 = ""
End Sub

Private Sub Command3_Click()
 On Local Error Resume Next
 Unload Me
End Sub

Private Sub Form_Load()
 On Local Error Resume Next
 'selecting the database location
 With CommonDialog1
     .DialogTitle = "PLEASE SELECT MYOCARDIAL(HEARTDISEASE) DATABASE"
     .Filter = "(*.mdb)|*.mdb"
     '.InitDir = CurDir
     .InitDir = "C:\HeartDisease"
     .filename = "HeartDisease.mdb"
     .Flags = cdlOFNExplorer Or cdlOFNFileMustExist Or cdlOFNPathMustExist
     .ShowOpen
     'make sure the filename is not an empty string
     If .filename <> "" Then
      'make sure that the database file returned is indeed biblio.mdb
      If Right(UCase(.filename), Len("HeartDisease.mdb")) = "HEARTDISEASE.MDB" Then
        gstrDBName = .filename
      Else
        MsgBox ("WARNING!!! INVALID DATABASE NAME ")
        End
      End If
      'Resume
```

```
        Else
          Unload Me
        End If
        'MsgBox gstrDBName
        'MsgBox "Lokasi Database Anda : -- > " & CurDir
        gstrRPTName = CurDir
       'Data1.DatabaseName = gstrDBName
      End With
      Data1.DatabaseName = gstrDBName
End Sub
```

FRMPENGGUNA1.FRM

```
Private Sub cmdAdd_Click()
  On Local Error Resume Next
  datPrimaryRS.Recordset.AddNew
End Sub

Private Sub cmdDelete_Click()
  On Local Error Resume Next
 msg = "Are you Sure to DELETE the Record ?"   ' Define message.
  Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
  Title = "WARNING : RECORD DELETION"  ' Define title.
  Response = MsgBox(msg, Style, Title)
  If Response = vbYes Then    ' User chose Yes.
    With datPrimaryRS.Recordset
     .Delete
     .MoveNext
     If .EOF Then .MoveLast
    End With
  End If
  datPrimaryRS.Refresh
End Sub

Private Sub cmdRefresh_Click()
  On Local Error Resume Next
  'This is only needed for multi user apps
  datPrimaryRS.Refresh
End Sub

Private Sub cmdUpdate_Click()
  On Local Error Resume Next
  datPrimaryRS.UpdateRecord
  datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified
End Sub

Private Sub cmdClose_Click()
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
  Unload Me
End Sub
```

```
Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
  On Local Error Resume Next
  'This is where you would put error handling code
  'If you want to ignore errors, comment out the next line
  'If you want to trap them, add code here to handle them
  MsgBox "Data error event hit err:" & Error$(DataErr)
  Response = 0  'Throw away the error
End Sub

Private Sub datPrimaryRS_Reposition()
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
  On Error Resume Next
  'This will display the current record position for dynasets and snapshots
  datPrimaryRS.Caption = "R: " & (datPrimaryRS.Recordset.AbsolutePosition + 1)
End Sub

Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
  On Local Error Resume Next
  If Save = True Then
  Style = vbYesNo + vbCritical + vbDefaultButton2
    ir = MsgBox("Data Have Been CHANGED, Are You SURE to SAVE CHANGES?", Style, "DATA
CHANGED " & Me.Caption)
    If ir = vbNo Then
      Save = False
      datPrimaryRS.UpdateControls
    End If
  End If
End Sub

Private Sub Form_Load()
  On Local Error Resume Next
  datPrimaryRS.DatabaseName = gstrDBName
End Sub

Private Sub Form_Unload(Cancel As Integer)
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
End Sub
```

FRMQUERY.FRM

```
Public selectx As Variant

Private Sub Command1_Click()
   On Local Error Resume Next
   Combo1.Visible = True
   Combo1.Text = "PICK CRITERIA"
End Sub
```

```
Private Sub Command2_Click()
  On Local Error Resume Next
  'print bill
   OutputDestination = 0
   CrystalReport1.Destination = OutputDestination
   fmlaText$ = selectx & ""
   MsgBox selectx
   CrystalReport1.SelectionFormula = fmlaText$
   If reportno = 1 Then
     CrystalReport1.ReportFileName = gstrRPTName & "\patient.rpt"
   End If
   If reportno = 2 Then
     CrystalReport1.ReportFileName = gstrRPTName & "\medical.rpt"
   End If
   'EXECUTE PRINT CALL
   On Error GoTo ErrorHandler
   CrystalReport1.Action = 1
    Exit Sub
 ErrorHandler:
   MsgBox CrystalReport1.LastErrorString
End Sub

Private Sub Command3_Click()
  On Local Error Resume Next
   Unload Me
End Sub

Private Sub Form_Load()
   On Local Error Resume Next
   Data1.DatabaseName = gstrDBName
   Text1 = "*"
   selectx = "{Patient.PatID} like '" & Text1 & "*'"
   Combo1.AddItem "Patient No"
   Combo1.AddItem "Patient Name"
   Combo1.AddItem "Patient NewIC"
   Combo1.AddItem "Patient OldIC"
   Combo1.AddItem "Sex"
   Combo1.AddItem "Ethnic"
   Combo1.AddItem "Site of Myocardial"
   Combo1.AddItem "Complication of Myocardial"
   Combo1.AddItem "Referral for Angiography"
End Sub

Private Sub Text5_Change()
      On Local Error Resume Next
      diga = Text5 & "*"
      If Combo1 = "Patient No" Then
         selectx = "{Patient.PatID} like '" & Text5 & "*'"
         Data1.RecordSource = "Select * from Patient WHERE PatID like '" & diga & "' order by PatID"
         Data1.Refresh
      End If
      If Combo1 = "Patient Name" Then
         selectx = "{Patient.PatName} like '" & Text5 & "*'"
         Data1.RecordSource = "Select * from Patient WHERE PatName like '" & diga & "' order by
PatName"
         Data1.Refresh
```

```
        End If
        If Combo1 = "Patient NewIC" Then
           selectx = "{Patient.NewIC} like '" & Text5 & "*'"
           Data1.RecordSource = "Select * from Patient WHERE NewIC like '" & diga & "' order by NewIC"
           Data1.Refresh
        End If
        If Combo1 = "Patient OldIC" Then
           selectx = "{Patient.OldIC} like '" & Text5 & "*'"
           Data1.RecordSource = "Select * from Patient WHERE OldIC like '" & diga & "' order by OldIC"
           Data1.Refresh
        End If
        If Combo1 = "Sex" Then
           selectx = "{Patient.Sex} like '" & Text5 & "*'"
           Data1.RecordSource = "Select * from Patient WHERE Sex like '" & diga & "' order by Sex"
           Data1.Refresh
        End If
        If Combo1 = "Ethnic" Then
           selectx = "{Patient.Ethnic} like '" & Text5 & "*'"
           Data1.RecordSource = "Select * from Patient WHERE Ethnic like '" & diga & "' order by Ethnic"
           Data1.Refresh
        End If
        If Combo1 = "Site of Myocardial" Then
           selectx = "{Patient.SiteMyo} like '" & Text5 & "*'"
           Data1.RecordSource = "Select * from Patient WHERE SiteMyo like '" & diga & "' order by
SiteMyo"
           Data1.Refresh
        End If
        If Combo1 = "Complication of Myocardial" Then
           selectx = "{Patient.CompCode} like '" & Text5 & "*'"
           Data1.RecordSource = "Select * from Patient WHERE CompCode like '" & diga & "' order by
CompCode"
           Data1.Refresh
        End If
        If Combo1 = "Referral for Angiography" Then
           selectx = "{Patient.Riskref} like '" & Text5 & "*'"
           Data1.RecordSource = "Select * from Patient WHERE Riskref like '" & diga & "' order by
Riskref"
           Data1.Refresh
        End If
End Sub
```

FRMSITE.FRM

```
Private Sub cmdAdd_Click()
 On Local Error Resume Next
 datPrimaryRS.Recordset.AddNew
End Sub
```

```
Private Sub cmdDelete_Click()
  On Local Error Resume Next
  msg = "Are you to DELETE the Record ?"   ' Define message.
  Style = vbYesNo + vbCritical + vbDefaultButton2 ' Define buttons.
  Title = "WARNING : RECORD DELETION"  ' Define title.
  Response = MsgBox(msg, Style, Title)
  If Response = vbYes Then    ' User chose Yes.
    With datPrimaryRS.Recordset
     .Delete
     .MoveNext
     If .EOF Then .MoveLast
    End With
  End If
End Sub

Private Sub cmdRefresh_Click()
  On Local Error Resume Next
  'This is only needed for multi user apps
  datPrimaryRS.Refresh
End Sub

Private Sub cmdUpdate_Click()
  On Local Error Resume Next
  datPrimaryRS.UpdateRecord
  datPrimaryRS.Recordset.Bookmark = datPrimaryRS.Recordset.LastModified
  DBGrid1.ReBind
End Sub

Private Sub cmdClose_Click()
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
  Unload Me
End Sub

Private Sub datPrimaryRS_Error(DataErr As Integer, Response As Integer)
    On Local Error Resume Next
  'This is where you would put error handling code
  'If you want to ignore errors, comment out the next line
  'If you want to trap them, add code here to handle them
  MsgBox "Data error event hit err:" & Error$(DataErr)
  Response = 0  'Throw away the error
End Sub

Private Sub datPrimaryRS_Reposition()
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
  On Error Resume Next
  'This will display the current record position for dynasets and snapshots
  datPrimaryRS.Caption = "R: " & (datPrimaryRS.Recordset.AbsolutePosition + 1)
End Sub
```

```
Private Sub datPrimaryRS_Validate(Action As Integer, Save As Integer)
  On Local Error Resume Next
  If Save = True Then
  Style = vbYesNo + vbCritical + vbDefaultButton2
    ir = MsgBox("Data Have Been CHANGED, Are You SURE to SAVE CHANGES?", Style, "DATA
CHANGED " & Me.Caption)
    If ir = vbNo Then
      Save = False
      datPrimaryRS.UpdateControls
    End If
  End If
End Sub

Private Sub Form_Load()
  On Local Error Resume Next
  datPrimaryRS.DatabaseName = gstrDBName
End Sub

Private Sub Form_Unload(Cancel As Integer)
  On Local Error Resume Next
  Screen.MousePointer = vbDefault
End Sub
```

MODULEMAIN.FRM (Main Module for MIMIS)

```
Public newrekod As Variant
Public penggunax As Variant
Public reportno As Variant
Public str As Variant

Public gstrDBName As String
Public gstrRPTName As String

Public NNMaxValueFile As Variant
Public NNResultFile As Variant
```

MODULEMLP001 (Main Module for Back Propagation Network)

```
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

'** DECLARE Global Var ****
Public TrainFileName As Variant       ' Training data file name
Public TestFileName As Variant        ' Testing data file name
Public ResultFileName As Variant      ' Result file name to store model weight

Public NumOfInputNode  As Integer     ' Number of inputs nodes
Public NumOfHiddenNode As Integer     ' Number of hidden nodes
Public EPSILON As Double              ' Maximum Mean Square Error to stop training
Public DELTA As Double                ' Learning rate
Public ALFA As Double                 ' Momentum rate
Public bin As Double                  ' Set the bin Number

Public testCOND As Integer            ' Testing condition
Public PercentCorrect As Integer      ' No of correct output
Public ArrayOfInput() As Double       ' Array of Input data
```

```
Public WtInputToHidden() As Double      ' Weight between input and hidden nodes
Public WtHiddenToOutput() As Double     ' Weight between hidden and output
Public ChgWtInputToHidden() As Double   ' Changes in weights used to adjust the weights
                            ' of next training patterns (between input nodes
                            ' and hidden nodes)
Public ChgWtHiddenToOutput() As Double  ' Changes in weights used to adjust the weights of
                            ' next patterns (between hidden and output nodes)
Public Z() As Double                ' Results of activation of inputs to hidden nodes
Public Y As Double                  ' Results of activation of hidden to output nodes
Public DesiredOutput As Double          ' Desired output of a training pattern
Public totaler As Double            ' Total square error calculated for all training

Public ehid() As Double             ' Error calculated for all hidden nodes
Public eout As Double               ' Error calculated for the output nodes
Public epoch As Integer             ' number of iteration
Public NumOfTrain As Integer            ' number of training patterns
Public NumOfTest As Integer             ' number of testing patterns
Public MaxEpoch As Integer            ' Number of Maximum Epoch

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function SetParameter()
'** This function will set the MLP parameter ***
'** and also redim the array             ***

  ' Get MLP parameters
  MaxEpoch = frmMLP002a.Text1
  NumOfInputNode = frmMLP002a.Text2
  NumOfHiddenNode = frmMLP002a.Text3
  EPSILON = frmMLP002a.Text4
  DELTA = frmMLP002a.Text5
  ALFA = frmMLP002a.Text6
  bin = frmMLP002a.Text11

  ' Initialize var
  testCOND = 0
  epoch = 0
  numest = 0

  ' Reinitialize Array
  ReDim ArrayOfInput(NumOfInputNode)
  ReDim WtInputToHidden(NumOfInputNode, NumOfHiddenNode)
  ReDim WtHiddenToOutput(NumOfHiddenNode)
  ReDim ChgWtInputToHidden(NumOfInputNode, NumOfHiddenNode)
  ReDim ChgWtHiddenToOutput(NumOfHiddenNode)
  ReDim Z(NumOfHiddenNode)
  ReDim ehid(NumOfHiddenNode)

End Function
```

```
'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function InitWeight() As Integer
'** This function will initialize the Input-To-Hidden Weight and Hidden-To-Output
'** Weight to a small random value between 0.5 t -0.5

   Dim i, j As Integer

   ' Uses the Randomize statement to initialize the random-number
   ' generator. Because the number argument has been omitted,
   ' Randomize uses the return value from the Timer function as the new seed value.


    Rnd (-1)
    Randomize (frmMLP002a.SeedNo)
   ' Initialize weight for input-hidden layer
   For i = 0 To NumOfInputNode - 1
      For j = 0 To NumOfHiddenNode - 1
         WtInputToHidden(i, j) = CDbl((0.5 * Rnd) + -0.5)
         ChgWtInputToHidden(i, j) = 0
      Next j
   Next i
   ' Initialize weight for hidden-output layer
   For i = 0 To NumOfHiddenNode - 1
      WtHiddenToOutput(i) = CDbl((0.5 * Rnd) + -0.5)
      ChgWtHiddenToOutput(i) = 0
   Next i
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function FeedForward()
   '-----------
   'FEEDFORWARD
   '-----------
   'step 3 : Each input unit (Xi, i = 1...n) receives input signal xi,
   '       and broadcasts this signal to all units in the layer above (the hidden unit)

   Dim i, j As Integer
   Dim Zin As Double ' Sum of Weight input to Hidden
   Dim Yin As Double ' Sum of Weight input to Output

   'Step 4 : Each hidden unit(Zj, j = 1...p) sums it weighted input signal

   For j = 0 To NumOfHiddenNode - 1
      Zin = 0
      For i = 0 To NumOfInputNode - 1
         Zin = Zin + ArrayOfInput(i) * WtInputToHidden(i, j)
      Next i
      'Step 4.1 : Applies its activation function to compute its output signal
      Z(j) = 1 / (1 + Exp(-Zin))
   Next j

   'Step 5 : Each output unit(Yk, k = 1...m) sums its weighted input signal
   Yin = 0
   For i = 0 To NumOfHiddenNode - 1
        Yin = Yin + Z(i) * WtHiddenToOutput(i)
```

```
    Next i
    'Step 5.1 : Applies its activation function to compute its output signal
    Y = 1 / (1 + Exp(-Yin))
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function PrintResult()

   If testCOND = 1 Then
      frmMLP002b.List1.AddItem "Record :" & Format(NumOfTest + 1, "0###") & " Target " &
DesiredOutput & "     NET : " & Format(Y, "0#.####")
   End If
   If testCOND = 2 Then
      frmMLP002b.List1.AddItem "Record : " & Format(NumOfTrain + 1, "0###") & " Target " &
DesiredOutput & "     NET : " & Format(Y, "0#.####")
   End If
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function BackProp()

   '------------------------
   'BACKPROPAGATION OF ERROR
   '------------------------
   'Step 6: Each output unit (Yk, k = 1..m) recieves a target
   '     pattern corresponding to it input training pattern,
   '     computes it error information term

   Dim i, j As Integer
   Dim temp As Double
   Dim e As Double

   ' Calculation of error
   e = 0
   eout = 0
   eout = DesiredOutput - Y

   ' When a training pattern is presented to the net, its total square
   ' error (E) will be summed up with the E of previous pattern
   NumOfTrain = NumOfTrain + 1
   e = 0.5 * (eout) * (eout)
   totaler = totaler + e

   ' error backpropated to the hidden layer
   For i = 0 To NumOfHiddenNode - 1
      ehid(i) = 0
   Next i

   For i = 0 To NumOfHiddenNode - 1
      ehid(i) = ehid(i) + WtHiddenToOutput(i) * Y * (1 - Y) * eout

   Next i
   ' adjust the weights of both layers using momentum
   For i = 0 To NumOfHiddenNode - 1
      temp = DELTA * Z(i) * Y * (1 - Y) * eout
      WtHiddenToOutput(i) = WtHiddenToOutput(i) + temp + ALFA * ChgWtHiddenToOutput(i)
```

```
      ChgWtHiddenToOutput(i) = temp
    Next i

    For i = 0 To NumOfInputNode - 1
      For j = 0 To NumOfHiddenNode - 1
        temp = DELTA * ArrayOfInput(i) * Z(j) * (1 - Z(j)) * ehid(j)
        WtInputToHidden(i, j) = WtInputToHidden(i, j) + temp + ALFA * ChgWtInputToHidden(i, j)
        ChgWtInputToHidden(i, j) = temp
      Next j
    Next i
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function CalPercentCorrect()

    NumOfTest = NumOfTest + 1
    NumOfTrain = NumOfTrain + 1
    mybin = 1 / bin
    If Abs(DesiredOutput - Y) < mybin Then
      PercentCorrect = PercentCorrect + 1
    End If
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function WriteModelWeight()

    Dim i, j As Integer

    Write #2, "Number of Input ", NumOfInputNode ' write no of input node
    Write #2, "Number of Hidden ", NumOfHiddenNode ' write no of hidden node
    'write Input-Hidden Weight
    For i = 0 To NumOfInputNode - 1
      For j = 0 To NumOfHiddenNode - 1
        Write #2, "Weight of input-hidden ", i, j, WtInputToHidden(i, j)
      Next j
    Next i
    ' Write Hidden-Output Weight
    For i = 0 To NumOfHiddenNode - 1
      Write #2, "Weight of hidden-output ", i, WtHiddenToOutput(i)
    Next i
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function training()

    Dim i As Integer
    ReDim ArrayOfInput(NumOfInputNode)

    Do
      epoch = epoch + 1
      NumOfTrain = 0
      totaler = 0

      'open file and read data
      Open TrainFileName For Input As #1
        Do
```

```
        'Read input
        For i = 0 To NumOfInputNode - 1
          Input #1, ArrayOfInput(i)
        Next i
        Input #1, DesiredOutput ' read target

        FeedForward
        BackProp

      Loop Until EOF(1) ' /*end of while*/
      Close #1
      frmMLP002a.Command3.Caption = Format(totaler / NumOfTrain, "#.######")
      frmMLP002a.Command2.Caption = epoch
    Loop Until ((totaler / NumOfTrain) < EPSILON Or epoch > MaxEpoch)
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function TestingResult()

  Dim i As Integer
  ReDim ArrayOfInput(NumOfInputNode)

  NumOfTest = 0
  testCOND = 1
  PercentCorrect = 0
  Open TestFileName For Input As #1
      Do
        For i = 0 To NumOfInputNode - 1
          Input #1, ArrayOfInput(i)
        Next i
        Input #1, DesiredOutput
        'call function
        FeedForward
        PrintResult
        CalPercentCorrect
      Loop Until EOF(1)

  Close #1

End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~`
Function TrainingResult()

  Dim j As Integer
  Dim t(), g As Double
  ReDim t(NumOfInputNode)
  ReDim ArrayOfInput(NumOfInputNode)

  testCOND = 2
  PercentCorrect = 0
  NumOfTrain = 0
  Open TrainFileName For Input As #1
      Do
        For i = 0 To NumOfInputNode - 1
          Input #1, ArrayOfInput(i)
```

```
            Next i
            Input #1, DesiredOutput
            'call function
            FeedForward
            PrintResult
            CalPercentCorrect
        Loop Until EOF(1)

         Close #1

End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
Function OpenTrainFile()

'selecting the database location
 With frmMLP002a.CommonDialog1
      .DialogTitle = "PLEASE SELECT TRAINING DATA"
      .Filter = "(*.txt)|*.txt"
      .InitDir = CurDir
      .filename = "AnginaTrain.txt"
      .Flags = cdlOFNExplorer Or cdlOFNFileMustExist Or cdlOFNPathMustExist
      .ShowOpen
      'make sure the filename is not an empty string
      If .filename <> "" Then
       'make sure that the database file returned is indeed biblio.mdb
        TrainFileName = .filename
        frmMLP002a.Text8 = TrainFileName
      Else
        MsgBox ("WARNING!!! INVALID DATA FILE NAME ")
      End If
    End With
End Function

'~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~

Function OpenTestFile()

   'selecting the Test data location
   With frmMLP002a.CommonDialog1
      .DialogTitle = "PLEASE SELECT TESTING DATA"
      .Filter = "(*.txt)|*.txt"
      .InitDir = CurDir
      .filename = "AnginaTest.txt"
      .Flags = cdlOFNExplorer Or cdlOFNFileMustExist Or cdlOFNPathMustExist
      .ShowOpen
      'make sure the filename is not an empty string
      If .filename <> "" Then
         TestFileName = .filename
         frmMLP002a.Text10 = TestFileName
      Else
         MsgBox ("WARNING!!! INVALID DATA FILE NAME ")
      End If
   End With
End Function
```