INTELLIGENT AGENT FOR ELECTRONIC COMMERCE USING NEURAL NETWORK APPROACH

by

YAP HUI SUN

A thesis presented to the Faculty of Computer Science and Information Technology of University of Malaya in partial fulfillment of the Requirements for the Degree of

BACHELOR OF COMPUTER SCIENCE

UNIVERSITY OF MALAYA

2000/2001

ACKNOWLEDGMENTS

Firstly, I would like to take this opportunity to express my appreciation to my thesis supervisor Mr. Woo Chaw Seng for his guidance and advices in the process of preparing the project. Secondly, I would like to thank Assoc. Prof. Dr. N. Selvanathan for being the moderator of the project.

I would like to thank my teammate who has done the Filtering agent part of the system designed in the project, Ms. Lai Siew Cheng, for her encouragement and cooperation in the process of preparing this project.

Last but not least, I wish to express my heartiest appreciation to my family and Mr. C.K. Cheng for their constant support and encouragement.

TABLE OF CONTENTS

TABLE OF CONTENTS (continued)

TABLE OF CONTENTS (continued)

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

This project is aims to developing a tool that able to collects relevant information from the World Wide Web and a tool that can be used to assists users in decision-making in the activities of e-commerce. It is been built by using the neural network of artificial intelligence approach. In more specific, the back propagation network was been used.

This report contains seven chapters that uncovered the details of the project from the beginning through the end of the system evaluation and a conclusion chapter. These chapters and its contain are been arranged as the following:

Chapter I - Introduction, introducing the project in various dimensions, which includes the briefing of project definition, objective and scope.

Chapter II - Literature Review, containing the finding on the areas related to the project, approach and algorithm used in the project and analysis of the existing system.

Chapter III – Methodology, justifying the method chosen for the system development, project planning that includes project schedule and system modules, procedure of the system, which are the functional and non-functional requirements, hardware and software used to develop the system.

Chapter IV – System Design, describing the system flow using graphic representation techniques and specifying in deep on both training and prediction module. It also describes on the database design and screen design of the system.

Chapter V – System Implementation and Testing, describing the system implementation of the development environment, development of webs, Prediction Agent and ActiveX component. It also describes the testing of the system.

Chapter VI – System Evaluation, presenting the evaluation of the system in terms of its strengths as well as the limitations.

Chapter VII – Conclusion, describing the problems encountered and solutions taken during the development of the system. Recommendations for future enhancements on the system and a conclusion that summarized of the project from the beginning of introduction through the end of system evaluation are included in this chapter.

CHAPTER I

INTRODUCTION

This introductory chapter gives a description or purpose of the project and problems to be solved. The significance and rationale of the project will be discussed here. Furthermore, the system functions, limitations and its assumptions will also be dwelt into later in this chapter.

1.1 Project Definition

The Internet provides a vast amount of information on virtually every conceivable topic. A rapidly growing segment of the Internet is Electronic Commerce. For shear convenience and preservation of time, consumers are looking for suppliers selling products and services on the Internet. Meanwhile, suppliers are looking for buyers to increase market share. It is usually a challenge for ordinary users to find exactly what they are looking for. In order to address this problem, a number of search engines have been created. Since the creation of search engine, the task of finding information on the Internet has become easier. Unfortunately, there are still some problems, which this tool cannot deal with. Besides that, search engines can be difficult to little experienced users that have to translate the query into a set of keywords linked through Boolean operators. Moreover, search engines rank the query results using criteria that can produce counter-intuitive results (for example using the frequency of query terms), which user is bound

to get some irrelevant information or in some cases information not related at all to what the user demanded, and force the user to scan almost all the documents retrieved. This task is very time-consuming and they may not find what they want.

The vast amount of information on the Internet and the weakness of search engines had caused a great deal of problems for both the consumer and the seller. These problems have led to the relatively slow growth of electronic commerce. It is speculated that intelligent agents will be able to solve these problems. Intelligent agents will be able to sort through the clutter on the Internet, resulting in the selection of specific brands, products, and stores. These entities will also be able to speed up the process of locating items on the Internet and leave users more time to do other, more productive or enjoyable tasks.

This project aims to build an intelligent agent, a tool that collects information from the web to assist users in decision-making. This tool consists of two intelligent sub-agents, which are filtering agent and prediction agent. Based on the domain of houses, the private residential properties, this agent had to carry out two main functions, which are to help the user to retrieve only relevant information required by the user and predict the price of the particular house as to assist user in decision-making. Artificial neural network approach will be used in developing this system in which back propagation is chosen as the learning algorithm for the agent. Therefore, a training module has to be provided to train the agent to become intelligent in doing its tasks. Besides of that, this tool also has some other relevant features that provide useful information to user in the

activity of e-commerce. These include a loan calculator and links to related fields in house purchasing.

1.2 Objective

The main objective of this project is to develop a tool that is intelligent enough to assist user in the activities of electronic commerce, which in this case, to collect relevant information on a particular house specified by user from the database and provide a price predictor as an assistant to user in price comparison and decision-making. This project also includes a training module that is responsible in providing training to the agent, as to increase the speed and accuracy of the agent in doing its tasks.

The housing development in Malaysia is progressing very fast especially for the last few years. Because of this, a lot of housing developers and property agents had put their web site on the Internet for the purpose of providing information about the houses they built to the users. Houses are built everywhere through out the country and it also come in various sizes and types. In order to search for a particular kind of houses, the user have to consider the location, types, sizes, prices, number of bedrooms and so on. This proved to be a time-consuming task especially if the user searches using search engines on the Internet because it will only returned the URL of the sites. They may have to browse through many web sites to get what they want and in some cases the sites provided is not relevant at all to what the user required. Even though the sites are relevant, user still has to go to these sites to compare the prices, location and so on.

This problem can be solved using the filter agent that is used to get relevant information from the web pages returned by the World Wide Web on behalf of a user. Users interact with this agent by submitting agent requests. The agent will get the user inputs and the data of houses store in database. These data will be sent to the network for filtering out the relevant data, which is required by the users. These relevant data will then be sent back to the database and will be displayed to the user.

A price predictor will be another intelligent tool, which make prediction of property's price based on user inputs. The predicted price (outcome) can be useful for the user to do price-comparison with the list of searching results and thus assist the user in decision-making. The rationale of this project is to build an intelligent agent for the above purpose.

Apart from that, this tool also provides other useful information to user such as loan calculator and links of related fields in house purchasing. It also provides a feedback form to get the comment from users as to further enhance the ability of the system.

1.3 Scope

The function of this system is to provide an agent to help the users to search for a particular house on the web, based on their requirement. Besides, it also provides the function of making price prediction of the user require house. This aims to help the user in decision-making based on the predicted prices of the specific house. The administrator will be given the authority in training and testing the prediction agent based on the data set selected.

This system has its limitations. The domain covered in this system is on houses in Malaysia. Therefore, the searching will be conducted on the houses available only in Malaysia. Price prediction will be based on four input vectors, which are location, type, bedroom number and size of the residential property. Due to this limitation, the prediction agent does not reflect a real predictor. Besides that, the system needs time to train the neural network embedded by using the training module. This module needs a large number of data sets to increase the accuracy of the output results.

With the functions provided, the output of the system will be a list of relevance description of a particular house the predicted price of the house. The user will be provided with a detail of performance as the outcome of training and testing the prediction agent.

This system will have the following features:

1) User interfaces include search form, predictor form and results page.

2) Training and testing module of prediction agent for administrator.

3) A loan calculator for calculation of monthly installment.

4) Others links that provide services related to house purchasing.

1.4 Summary

This chapter gives an overview of the project that includes the aims, relevance, significance and the scope of the development of an intelligent agent for e-commerce. Besides that, it also covers the importance and feasibility of the project as well as the project limitations and expected outcomes.

CHAPTER II

LITERATURE REVIEW

This chapter will discuss about the facts finding techniques, meaning and overview of areas covered in the project literally. The major areas that will be covered are as follow:

- Artificial Neural Network
- Intelligent Agent
- Electronic Commerce
- Artificial Intelligence in Electronic Commerce

2.1 Facts Finding Techniques

Useful information and recommendation are obtained through carrying some efforts of the finding. The following shows the techniques used for this project:

2.1.1 Internet surfing

Surfing the Internet is indeed a good way of finding information. There are many rich and broad-based Web pages providing a lot of information on electronic commerce. Feedback from some Frequent Asked Questions (FAQ) sessions helped clear up misconceptions or erroneous plans that had been formed. Online tutorials regarding certain programming language can also be obtained through this method of

surfing net. Existing intelligent agent systems and e-commerce sited also help in giving ideas on the features of the system development.

2.1.2 Research

All the research works are approached from the point of view of this system, which involved reviewing academic materials that contain relevant information, especially on some existing technical works.

2.2 Artificial Neural Network (ANN)

2.2.1 Introduction

The fundamental processing element of a neural network is a neuron. This building block of human awareness encompasses a few general capabilities. Much is still unknown about how the brain trains itself to process information, so theories abound. In the human brain, a typical neuron collects signals from others through a host of fine structures called dendrites. The neuron sends out spikes of electrical activity through a long, thin stand known as an axon, which splits into thousands of branches. At the end of each branch, a structure called a synapse converts the activity from the axon into electrical effects that inhibit or excite the connected neurons. When a neuron receives excitatory input that is sufficiently large compared with its inhibitory input, it sends a spike of electrical activity down its axon. Learning occurs by changing the effectiveness of the synapses so that the influence of one neuron on another changes. Figure 2.1 below show a biological neuron in a human brain.

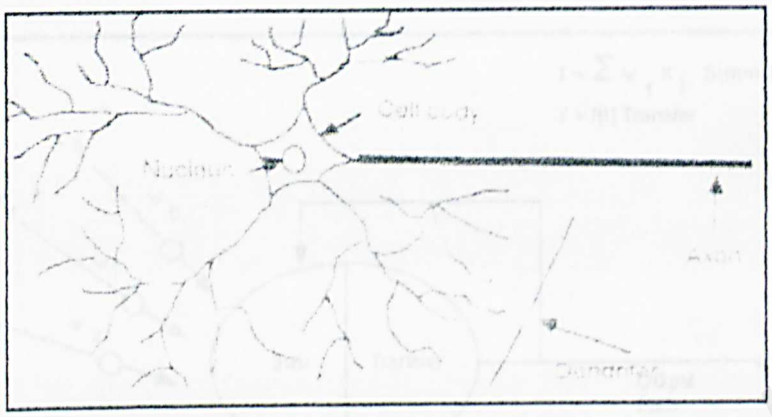

Figure 2.1 A biological neuron

Artificial neural networks are parallel computational models comprised of densely interconnected adaptive processing units. These networks are fine-grained parallel implementations of nonlinear static or dynamic systems. A very important feature of these networks is their adaptive nature, where "learning by example" replaces "programming " in solving problem [1].

The term "neural network" describes a class of models, which appear under different names in the literature: neural networks, neural computation, artificial neural systems, connectionist models and parallel-distributed models [2].

This artificial neural network is comprised of various neurons connected by its weight. These neurons are modeled after the biological neuron shown above. Figure 2.2 show the architecture of a basic artificial neuron.

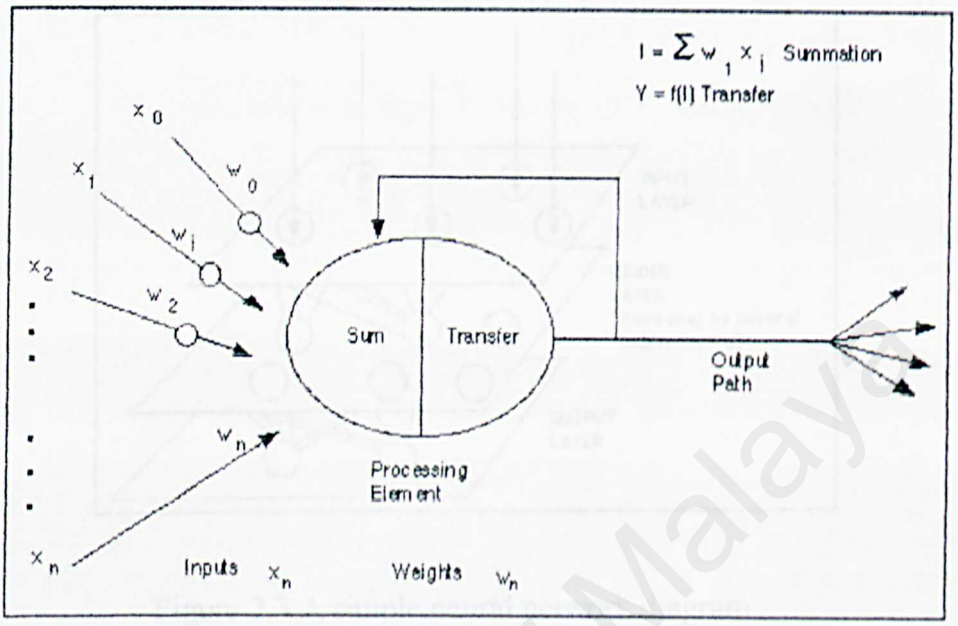

Figure 2.2 A basic artificial neuron

An artificial neuron is a basic processing element of a neural network. Refer to the above figure the various inputs to the network are represented by X (n). This input is analog to the dendrites of a biological neuron. Each of this input is multiplied by a connection weight W (n). This product will be summed together and send to a transfer function to generate a result. There are many different transfer functions, which can be used such as hard limit, pure linear, log-sigmoid, etc.

The neural network can contain a lot of layers. A single layer is combined of one or more artificial neurons. Figure 2.3 shown a simple neural network with three layers namely input layer, hidden layer and output layer.

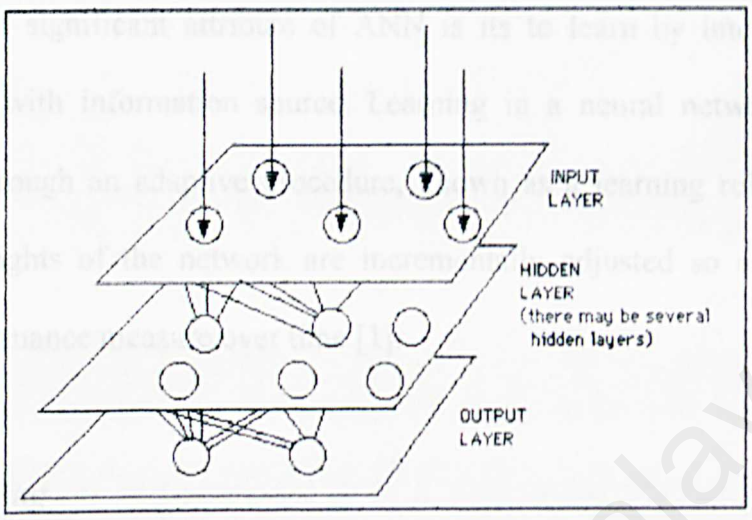

Figure 2.3 A simple neural network diagram

2.2.2 Classification of ANN

A neural network can be classified by its method of determining the weights on the connections, called its learning algorithm, and its pattern of connections between the neurons, call its network topology. Basically, there are 2 major learning algorithms in ANN, which are supervised learning and unsupervised learning. Network topologies in ANN are divided in two categories, which consist of feed forward, and feedback network. Each of this learning methods and network topologies will be discussed below.

2.2.2.1 Learning Algorithm

One of the most significant attribute of ANN is its to learn by interacting with its environment or with information source. Learning in a neural network is normally accomplished through an adaptive procedure, known as a learning rule or algorithm, whereby the weights of the network are incrementally adjusted so as to improve a predefined performance measure over time [1].

Supervised Learning

In supervised learning, the network is trained on a training set consisting of vector pairs. One vector is applied to the input of the network; the other is used as a "target" representing the desired output.

Unsupervised Learning

Unsupervised learning, sometimes called self-organization requires only input vectors to train the network. During the training process the network weights are adjusted so that similar inputs produce similar output. This is accomplished by the training algorithm that extracts statistical regularities from the training set, representing them as the values of network weights [3].

2.2.2.2 Network Topology

There are two major kinds of network topology, which are feed forward, and feedback neural network.

Feed forward neural network

In a feed forward neural network, the connections between units do not form cycles. Feed forward neural networks usually produce a response to an input quickly. Most feed forward neural networks can be trained using a wide variety of efficient conventional numerical methods in addition to algorithms invented by neural network researchers.

Feedback neural network

In a feedback or recurrent neural network, there are cycles in the connections. In some feedback neural networks, each time an input is presented, the neural network must iterate for a potentially long time before it produces a response. Feedback neural networks are usually more difficult to train than feed forward neural networks.

In this project, back propagation method will be used in which it uses supervised learning and it is a feed forward network.

2.2.3 Back Propagation Neural Network

The back propagation network is a very popular model in neural network. It does not have feedback connections, but errors are back propagated during training. Least mean squared error is used. Many applications can be formulated for using back propagation network and the methodology has been a model for most multi-layer model.

Back propagation network has a gradient descent method to minimize the total squared error of the output computed by the neural network. The training of the neural network by back propagation involves three stages: the feed forward of the output training pattern, the calculation and back propagation of the associated error, and the adjustment of the weight. After training, application of the neural network involves only the computations of the feed forward phase. Even if training is slow, a trained neural network can produce its output very rapidly.

Algorithm of back propagation network

Three stages of training:

During feed forward, each input unit receives an input signal and broadcasts the signal to each of the hidden units. Each hidden units then computes its activation and sends its signal (Z_j) to each output unit. Each output unit computes its activation (Y_k) to form the corresponding of the neural network for the given input pattern.

During training, each of the output unit compares its computed activation Y_k with its targets value T_k to determine the associated error for that pattern with that unit. Based on this error, the factor Delta $_k$ (k=1, 2, .., m) is computed. Delta $_k$ is used to distribute

the error, at output unit back to all units in the previous layer. In a similar manner, Delta $_j$ is computed for each hidden unit. It is not necessary to propagate the error back to the input layer, but Delta $_j$ is used to update the weights between the hidden layer and the input layer.

After all of the s factors have been determined, the weights for all layers are adjusted simultaneously. The adjustment to the weight W_{jk} is based on the factor Delta $_k$ and the activation of the hidden unit. The adjustment to the weight V_{ij} is based on the factor Delta $_j$ and the activation of the input unit.

The activation function

The activation function is a non-linear function that, when applied to the net input of a neuron, determines the output of that neuron. Its domain must generally be all real numbers, as there is no theoretical limit to what the net input can be. However, in practice we can easily limit the net input by limiting the weights. An activation function for a back propagation net should have several important characteristics: it should be continuous, differentiable, and monotonically non-decreasing. Further more, for computational efficiency, it is desirable that its derivative be easy to compute. Usually, the function is expected to saturate, e.g. approach finite maximum and minimum values asymptotically.

2.2.4 Neural Network Programs

Neural network programs are a new kind of computing tool, which simulate the structure and operation of the human brain. They mimic many of the brain's most powerful abilities, including pattern recognition, association, and the ability to generalize by observing examples. Neural networks create their own model of the problem through a training process, so no programming is required. A trained network provides answers with lightning speed, in less than a second. User can retrain a network to use new, updated information in minutes.

Common uses for neural networks include medical diagnostic systems, insurance claim evaluations, sports event predictions, loan risk evaluations, pattern recognition, and business analysis and decision-making [4].

2.3 Intelligent Agent

There are several definitions of agents. One can also describe rather than define agents in terms of their task, autonomy, and communication capabilities. Some of the major definitions and descriptions of agents are:

i. Agents are semi-autonomous computer programs that intelligently assist the user with computer applications. Agents employ artificial intelligence techniques to assist users with daily computer tasks, such as reading electronic mail, maintaining a calendar, and filing information. Agents learn through example-based reasoning and are able to improve their performance over time.

ii. Agents are computational systems that inhabit some complex, dynamic environment. They sense and act autonomously in this environment. By doing so, they realize a set of goals or tasks.

iii. Agents are software robots. They can think and will act on behalf of a user to carry out tasks. Agents will help meet the growing need for more functional, flexible, and personal computing and telecommunications systems. Uses for intelligent agents include self-contained tasks, operating semi-autonomously, and communication between the user and systems resources.

The definition and description of an agent for this research are: Agents are software programs that implement user delegation. Agents manage complexity, support user mobility, and lower the entry level for new users. Agents are a design model similar to client-server computing, rather than strictly a technology, program, or product [5].

An intelligent agent is a computer system that is capable of flexible autonomous action in order to meet its design objectives. By flexible, it is meant that the system must be:

- **Responsive:** agents should perceive their environment (which may be the physical world, a user, a collection of agents, the Internet, etc.) and respond in a timely fashion to changes that occur in it.
- **Proactive:** agents should not simply act in response to their environment, they should be able to exhibit opportunistic, goal-directed behavior and take the initiative where appropriate.

- **Social:** agents should be able to interact, when they deem appropriate, with other artificial agents and humans in order to complete their own problem solving and to help others with their activities.

An agent can be applied in many domains such as in industrial application, medical application, commercial application, entertainment and so on. At a first glance, an agent seems to be the same as a search engine like Yahoo, AltaVista, Lycos, etc, but this is not true. The key difference though is that an agent is more interactive and can accomplish many tasks at many different locations. For example, if user searches on a keyword using a search engine such as Yahoo, the user will get a list of matches. The user can then follows those links and possibly get the information. If an agent was been used, on the other hand, the agent could submit entered keywords to many search engines, follow the corresponding links, and gather the information, all without any intervention from the user.

2.4 Electronic Commerce

Electronic commerce is defined as any form of business or administrative transaction or information exchange that is executed using any information and communications technology (ICT). Following this definition, information on these pages includes the latest technologies for enabling trade over the internet, the more established practices of conventional EDI (electronic data interchange) and bar coding, as well as the business reasons for implementation of these technologies [6].

Electronic Commerce is the buying and selling of goods and services across the Internet. Over 20 million people are currently buying products and services through the Internet. In 1998, online retail shoppers are expected to spend $ 4.8 billion, and by 2001, on-line retail sales are expected to top $17.8 billion. Currently, the frequent online shopper spends an average $672 per year on-line and this is expected to grow to $773 per year by 2001. Consumers are gaining confidence in the security of the Internet, which will further increase the growth in sales.

An e-commerce site can be as simple as a catalog page with a phone number, or it can range all the way to a real-time credit card processing site where customers can purchase downloadable goods and receive them on the spot. Electronic commerce merchants can range from the small business with a few items for sale all the way to a large on-line retailer such as Amazon.com.

2.5 Artificial Intelligence in E-commerce

Ever since the growth of e-commerce, shopping online has become a popular way to buy things. Consumers have to browse a lot of sites to get the lowest price on the product and it is very time consuming. Consumers want to be able to find a broad selection when looking for a product; however, the selection must be edited down to reasonable and manageable proportions. This editing feature is the job of the intelligent agent.

Recent significant progress in AI for electronic commerce includes:

- Practical shopping agents, including Web portal services that use knowledge representation, decision analysis, machine learning, and information retrieval technique.
- Practical recommender services, e.g., e-storefronts that use collaborative filtering.
- Practical data mining by sellers to learn customer-buying patterns.
- Practical customer-service help, including agent techniques to categorize and route e-mail, do case-based associative retrieval and make suggestions.
- Theory of economic decision-making, markets, negotiations, and contracts, including from the viewpoints of resource-bounded intelligence, game theory, distributed AI, negotiation, probabilistic and uncertain reasoning, and decision analysis.
- The theory and practice of auctions.
- Agent communication languages, including negotiation languages and protocols and knowledge interchange and the use of XML-encoded domain ontology and communication languages.
- Web information retrieval and information integration, including using NLP, text analysis and machine learning.
- Online product/service catalogs, e.g., techniques to aggregate catalogs.

Shopping agent is one kind of intelligent software agents, who is specialized in helping network users in the electronic commerce for information finding, shopping, and price-comparison. Even the secure electronic transaction can be fully handled by shopping agents. Currently, there are many shopping agents available for use. Below are few of the popular shopping agents on the web and their uses.

i. Fido the Shopping Doggie

Fido searches the Continuum Software database of merchant pages and produces a list of product descriptions and prices.

ii. Bargain Finder

Bargain Finder allows user to shop for the lowest music CD price via the sites it searches. User just type in the name of the CD user want to buy, and the "agent" searches specific music sites that have allowed their documents to be indexed.

iii. Firefly

Firefly also helps the user find music, but it is more sophisticated in that it learns the user's musical preferences. Firefly asks users to rate a number of different music artists, then it suggests other types of music that the user might like. The suggestion is based on correlations with what other people say they enjoy listening to, versus using artificial intelligence.

2.6 Summary

This chapter gives brief explanation on the topics researched and studied that are relevant to the project. Among the topics are artificial neural network, concepts of intelligent agent, electronic commerce, and the application of artificial intelligence in e-commerce, where the techniques of Internet surfing and research were been applied. These topics understanding will be followed by project analysis stage that comes after this chapter.

CHAPTER III

METHODOLOGY

The objectives specified in the first chapter were achieved using the algorithm in this chapter.

3.1 The Waterfall Model with Prototyping

Conventional development and prototyping may be combined. The waterfall model with prototyping is chosen because the strengths of each can be achieved on a single project. This model is actually the classic waterfall model combined with the prototyping approach in its early stages. The model is shown in Figure 3.1.

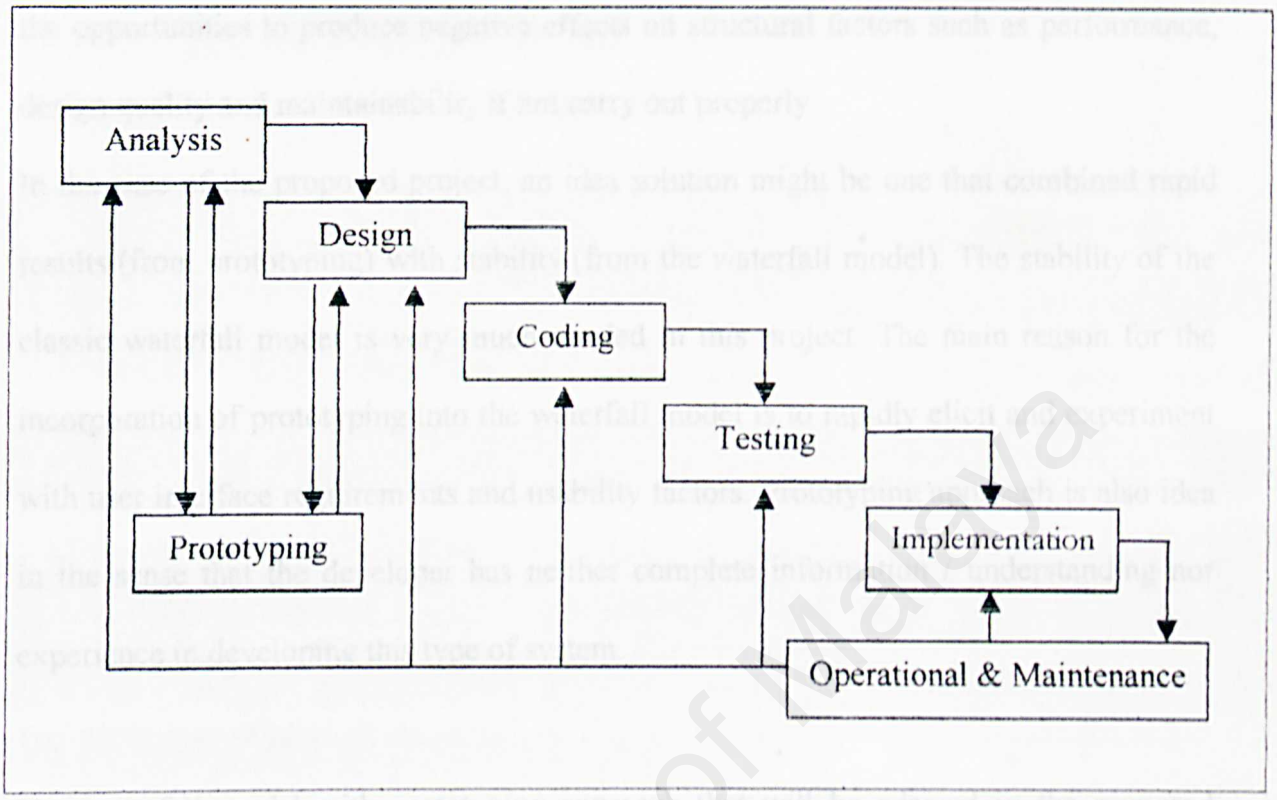

Figure 3.1 The waterfall model with prototyping

In the proposed development strategy, the waterfall model will serve as the base for the development because the steps of this model are very similar to the generic steps of software development process that are applicable to all software engineering paradigms. It also provides a template into which methods for analysis, design, coding, testing and maintenance can be placed. Prototyping will be involved in the early stages of the waterfall model where there is a need for experimentation and learning before commitment of any resources to develop the full-scale system. Prototyping will not be involved in the later stages of the development because its major drawback in increasing

the opportunities to produce negative effects on structural factors such as performance, design quality and maintainability if not carry out properly.

In the case of the proposed project, an idea solution might be one that combined rapid results (from prototyping) with stability (from the waterfall model). The stability of the classic waterfall model is very much needed in this project. The main reason for the incorporation of prototyping into the waterfall model is to rapidly elicit and experiment with user interface requirements and usability factors. Prototyping approach is also idea in the sense that the developer has neither complete information / understanding nor experience in developing this type of system.

The waterfall model with prototyping approach that will be adapted in the proposed project encompasses the activities at system analysis, system design, coding, testing and implementation.

Analysis

The goal to the system analysis is to understand the proposed system and to establish system requirements. The system analysis phase is concerned with data gathering and data analysis.

Data will be gathered from sources like written materials, Internet as well as observation and examination of others intelligent agent systems. The iterative process of prototyping-revision will be Data Flow Diagram (DFD) is chosen to analyze the collected data because it enables the information domain and functional domain to be

modeled at the same time. It is to be used to graphically show the flow of the data through the system.

The most important outcome from this phase will be an accurate system requirements specification.

Design

The system design phase is the phase in which requirements produced in the previous phase are translated into a representation of the system. This phase will be concerned with user interface design, database design and system design.

The interactive process of prototyping-revision will be used to revise the design of the user interface. Interface prototypes will be built using Microsoft Visual InterDev. Entity-relational (E-R) modeling will be involved in the logical design of the Microsoft Access database. In system design, structure chart will be involved in structuring the system's modules and flow chart might be used to depict the design of procedural details.

Coding

This stage translates and implements the detail design representation of the system into programming realization. Scripting languages such as VBScript and HTML might be used in coding the information and functional domain as well as the control of the proposed system. Microsoft Visual InterDev 6.0 is the proposed web-authoring tool that will be used to create web pages while MATLAB 5.3 for neural network development.

Microsoft Visual Basic 6.0 will be the tool to create ActiveX components and Microsoft Access 97 will be used to develop the database of the system.

Testing

Testing will be a critical step in assuring the quality of the developed system and will represent the ultimate review of specification, design and coding. First, unit testing will be performed to verify each program module. Next, integration testing is performed. It is to integrate unit-tested program modules and conduct tests that uncover errors associated with the interfacing of those modules. Validation test succeeds when the system functions in a manner that is reasonably expected.

Implementation

The finally stage of the development will be system implementation. The system will be implemented on its target software and hardware requirement. The whole system will be revise to uncover the necessity to add further enhancements. Maintenance process should be an ongoing activity in real development projects. Monitoring a necessary adjustments continue so that the system produces the expected results. However, system enhancements and maintenance will only be carried out in the proposed project if time constraint allowed.

3.2 Planning

3.2.1 Project Schedule

The project was done step by step according to the schedule shown in Table 3.1, which had been set at the beginning of the project.

Table 3.1 Project schedule

Key Activity	June	July	Aug	Sept	Oct	Nov	Dec	Jan
	00	00	00	00	00	00	00	01
Literature Review								
Requirement Analysis								
System Design								
Coding / Prototyping								
SystemTesting								
Documentation								

3.2.2 System Modules

The system consists of four modules, which are display module, filtering module, training module and prediction module.

Display module

This module contains the user interfaces, which is consists of two main screens:

i. User input screens

- Search form to get query from user to help them to obtain the required information.
- Feedback form for user to give comment on the system.

ii. Results screens

- Present a listing of relevant houses description.
- Present a predicted price to assist user in decision-making.
- Present the training and testing performance of the prediction agent.

Filtering module

This module will use neural network to do the filtering of the data in the database. It will filter out the relevant information, which is required by user from the database and display it to users.

Training module

This module will able the administrator to train and test the prediction neural network with sets of sample data. 150 data will be use as training data set to train the agent. Administrator can then test the agent to determine the accuracy of the agent in doing its task by using test data set provided.

Prediction module

This module will predict the price of user query house. The price will be determined by using back propagation neural network with four input criteria selected by the user through the input screen. These fields are state location, type, bedroom number and built-up area of the house.

3.3 Procedure

3.3.1 Functional Requirements

3.3.1.1 General user section

This section is responsible to communicate with user in getting information from user to proceed the searching, price prediction and train the network, and displays the outcome to user.

Display module

This is a front-end design with is responsible for the interactions between user and the system. It consists of the following:

i. Property search form

This form enables user to input his/her required criteria such as location, type, size and, number of bedroom of a house.

ii. Price prediction form

This form will gather the input of property detail from the user and send to prediction agent for price prediction.

iii. Property search result page

This page includes a listing of entire search results, which are the relevance houses description.

iv. Price prediction result page

This page displays a recommendation of predicted house price.

vi. Training result page

The training result of performance goal and prediction accuracy will be shown in this page.

vii. Testing prediction result page

This page displays the testing result in form of table together with the prediction accuracy achieved in testing process.

viii. Sending feedback form

User can gives comment on the system by using the feedback form provided. The collected comments will be used to improve the functionality of the system.

3.3.1.2 Agent section

There are two agents in this section, which are playing different roles in searching the most relevant information for the user.

Filtering module

This module aims to ensure that only relevant houses descriptions are proposed to the user. The filtering agent is responsible to distil the Web pages and extract relevant information.

Prediction module

The purpose of this module is to assist the user in decision-making by predicting the value of house. User's query terms are been used in the module to predict the selling price of a house.

3.3.1.3 Administration section

This section contains two modules that play an important role in system enhancement.

Training module

This module aims to train the prediction agent in producing the best results. It is also aims to test the degree of accuracy of the agent in doing its task. Two sets of data are required in this module, which are training data and test data.

Feedback administer

The administrator is responsible on collecting user feedback, which may be useful for system improvement and enhancement in the future.

3.3.1.4Others services section

This section provides links to several useful sites that are related on house purchasing. These links include house loan, insurance, legal firm and so on.

3.3.2 Non-Functional Requirements

i. Maintainability

The system are easy to modify and test in updating process to meet the new request, correcting errors, or move to a different computer system.

ii. Reliability

The system should maintain high reliability of data to reduce the failure at the prediction module when the output is predicted. The system should also operates in a user-acceptable manner when used in the environment for which it was designed, which does not produce dangerous or costly failure when it is applied in a reasonable manner.

iii. Efficiency

Implementation of the system corresponds to the most cost-effective computing resource utilization, where process that can be called or accessed in an unlimited number of times to produce similar outcomes at a creditable pace or speed.

iv. User friendliness

The design of the system and its interface should be user friendly and easy understanding by all level of the Internet users. Generally, the design of all the interfaces should conform to the following criterions:

-Consistent, in terms of screen design and error messages displayed.

-High degree of understandability and avoid memorization of events and commands.

v. Simplicity

Forms and screens are kept properly uncluttered in a manner that focuses the user attention.

vi. Understandability

Coding method used, allows other programmer to understand the logic of the program flows.

3.3.3 Hardware Used

The listing below shows the capability of hardware that is used to develop the system.

i. 200MHz Processor
ii. 96MB of RAM
iii. 2.1GB of Hard Disk
iv. VGA colour monitor
v. Keyboard
vi. Mouse

3.3.4 Software Used

The following are the software chosen to develop the system.

3.3.4.1 Hyper Text Markup Language (HTML)

The Hyper Text Markup Language, or HTML, is the language used to prepare Web hypertext documents. HTML contains commands, called *elements* or *tags*, to mark text as headings, paragraphs, lists, quotations, and so on. It also has tags for including images within the documents, for including fill-in forms that accept user input, and, most importantly, for including hypertext links connecting the document being read to other documents or Internet resources [7].

3.3.4.2 Active Server Pages (ASP)

ASP is the script that runs on the Microsoft Web Server. Its function is to generate HTML scripts for the client browser. Fast execution is one of the major reasons that ASP is chosen to develop the system. Time saving is really important to customer. Compared to CGI, ASP is easier to be used and is more flexible in changing codes as no compilation is involved. ASP enables dynamic Web design effortlessly. This feature makes the Web applications easy to maintain and modify to meet the new needs and requirements [8]. It is therefore selected as the main development tool for the server run script.

3.3.4.3 VB Script

VB Script is an interpreted script language from Microsoft. It is a subset of the Visual Basic programming language. Its function is to make the Web application more dynamic. VB Script will be embedded in the HTML pages to build the Web applications for the system. It is chosen due to its ease to learn and write program if compare to Java [9]. Although VBScript is still new if compare to the JavaScript, however it proved more reliable in working with the ASP. Besides, VB Script is a fast, portable, lightweight interpreter for use in World Wide Web browser and other Web applications.

3.3.4.4 Microsoft Access

Microsoft Access was designed to be a relational database management system with added features to its popular database tool to take advantage of the Internet. It contains tool sets that provide basic web database functionality. Besides, it is a tool that many people already own and feel comfortable with [8]. Due to these reasons, it had been chosen as the database type of the system.

3.3.4.5 MATLAB Neural Network Toolbox (NNT)

The MATLAB Neural Network Toolbox (NNT) is an all-purpose neural network environment. Everything is included, and most of it has someway been incorporated in the network object. It is a powerful tool in creating a neural network. High-level network creation functions are included in the toolbox [10]. Therefore, the software was been selected to develop the neural networks embedded in the system.

3.4 Summary

The waterfall model with prototyping was been chosen as the system development strategy due to its suitability mentioned above. Project time line was planned and system modules were been determined during the project planning. System procedure with functional and non-functional requirements was also included in this chapter. System design will come next as in the following chapter after the planning of system methodology.

CHAPTER IV

SYSTEM DESIGN

This chapter will show a clearer picture on the system design, which includes the overall system flow diagram, database design and screen design. It also uncovers a detail description of Training Module and Prediction Module structures.

4.1 System Overview

The system consists of five modules, which are:

i. Display module

ii. Filtering module

iii. Training module

iv. Prediction module

4.1.1 System implementation steps

The Intelligent Agent System was designed to helps user to obtain the required house information from the World Wide Web and propose a predicted price for the specified house to assists user in decision-making. The implementation steps includes the following chain of events:

- Solicit a subject from user query,
- Sent the user inputs and the houses information stored in database to the network,
- Filter out irrelevant Web pages and select relevant information,

- Rearrange the selected results,
- Predict the price for the query subject,
- Display the results to user.

Once the user fill in and submit the search form, the house criteria will be uploaded to the Search Agent. This in turn will send the user inputs and the house information stored in database to the neural network, which is the Filtering Agent. The Filtering Agent, with embed neural network, given a set of weight, will attempted to map the query terms (user input), which is represent as the input vectors to the output vector, which could indicate the "relevance" of the input vector to a particular house required.

The relevant Web pages will be select by the Filtering Agent as the output. On the other hand, the Prediction Agent, another neural network, which will predict the value of house required by user. In this case, the house price is the target output, and criteria such as location, house type, built-up area and number of bedroom are the variables used to predict the selling price. Before doing the prediction, the agent should be train and test by sets of training data and testing data respectively. After this network is trained, it could predict a selling price for a specific house. This predicted price is used to assist user in decision-making. Finally, the relevant information that was filtered out previously and the predicted price will be displayed to user.

This system is operates as the following sequence of flow as shown in Figure 4.1and Figure 4.2.

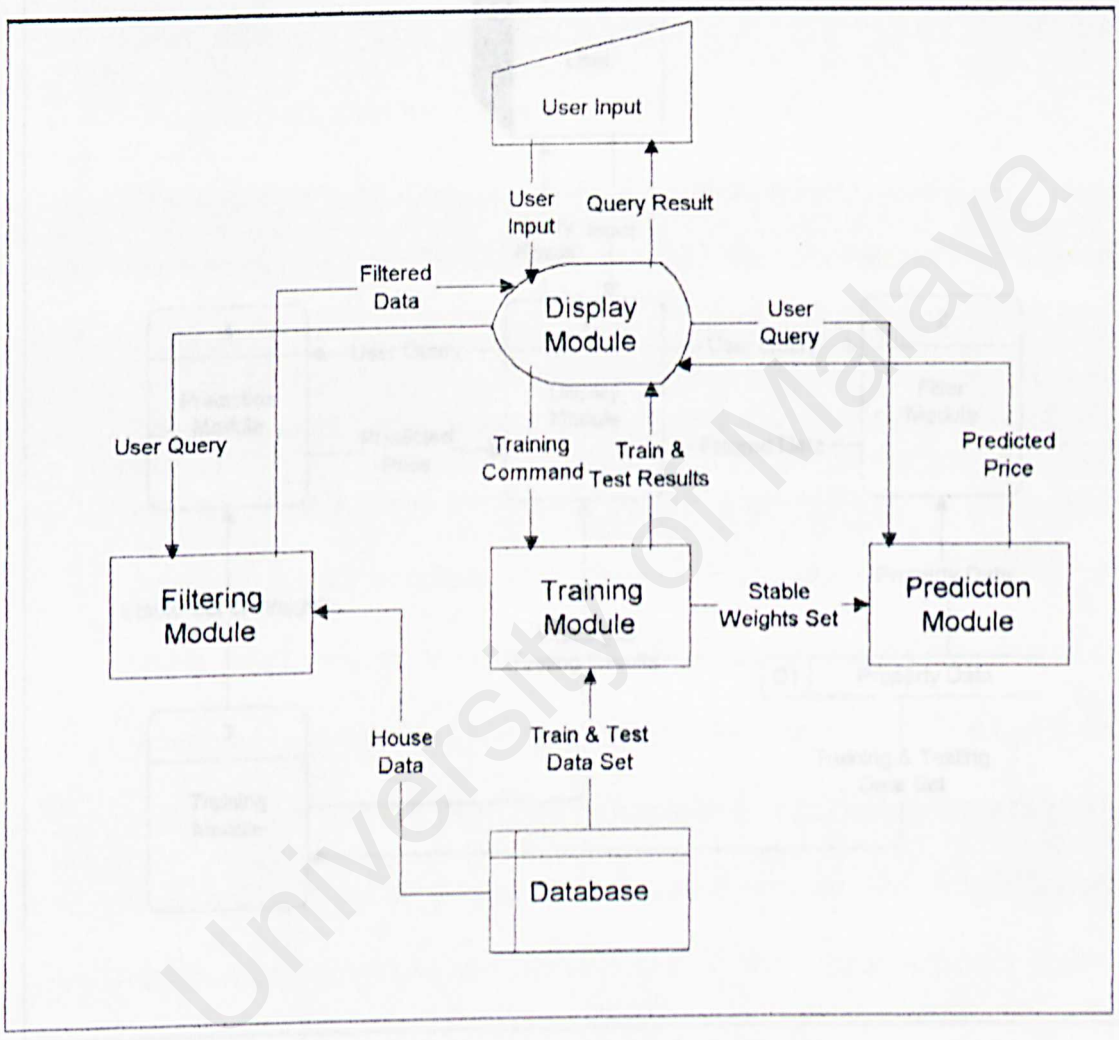

Figure 4.1System flow chart

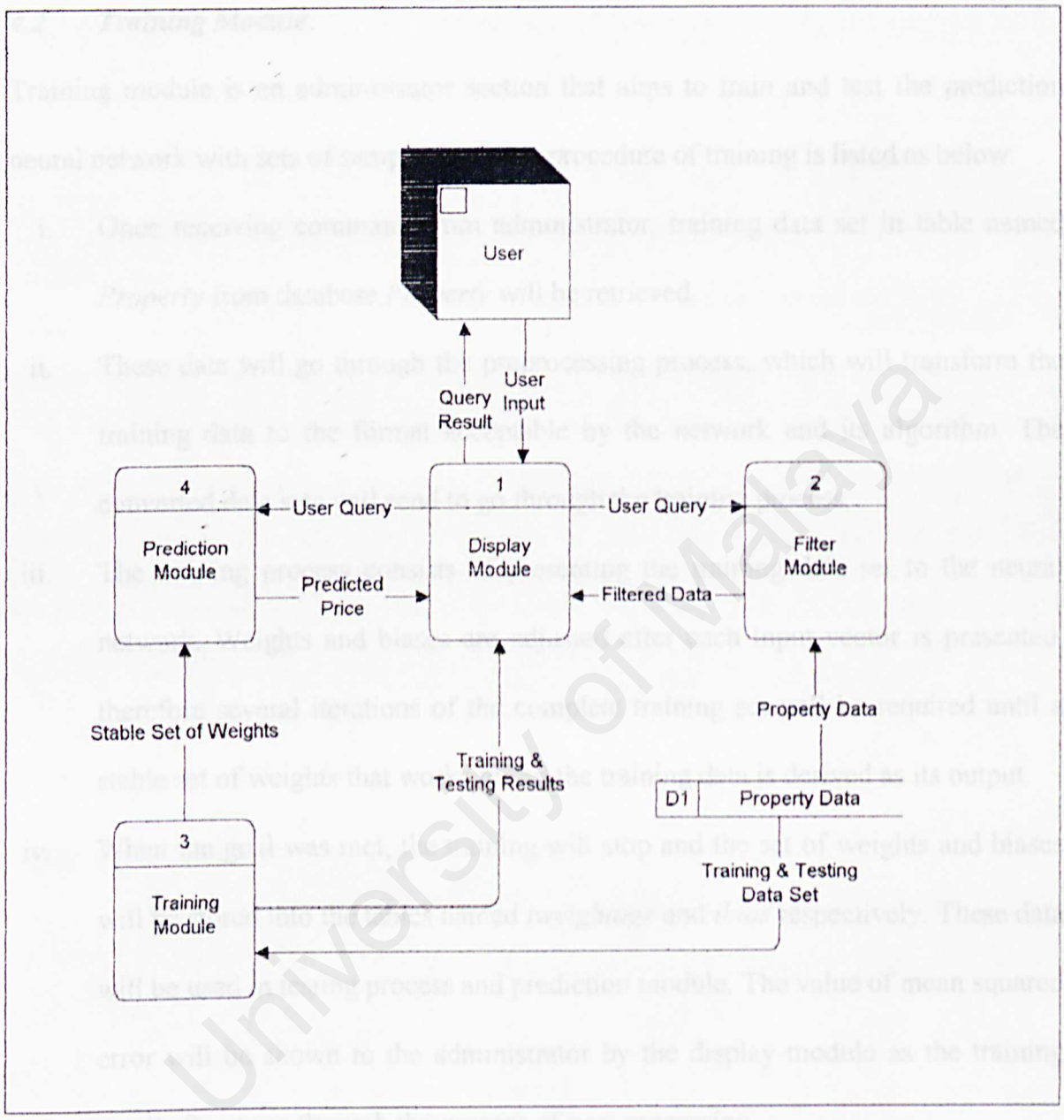

Figure 4.2 System data flow diagram

4.2 *Training Module*

Training module is an administrator section that aims to train and test the prediction neural network with sets of sample data. The procedure of training is listed as below:

i. Once receiving command from administrator, training data set in table named *Property* from database *Property* will be retrieved.

ii. These data will go through the preprocessing process, which will transform the training data to the format acceptable by the network and its algorithm. The converted data sets will send to go through the training process.

iii. The training process consists of presenting the training data set to the neural network. Weights and biases are adjusted after each input vector is presented, therefore several iterations of the complete training set will be required until a stable set of weights that works for all the training data is derived as its output.

iv. When the goal was met, the training will stop and the set of weights and biases will be stored into the tables named *iweightage* and *ibias* respectively. These data will be used in testing process and prediction module. The value of mean squared error will be shown to the administrator by the display module as the training result after gone through the process of post processing.

The testing process examines the performance of the network using the derived stable set of weights and biases from the training process, by measuring the ability of the network to classify the test data correctly. The procedure of testing is listed as below:

i. Administrator can test the agent after the training process. The data from table named *testset* will be used as the testing data that are new to the agent.

ii. The data will be preprocessed to the format acceptable by the neural network before start testing. The testing then begins by presenting the testing data to the neural network.

iii. The process output will be the mean squared error and a table of test data set together with difference between the desired price and predicted price. This output will be shown by the display module after being post processed.

The procedure is shown in the data flow diagram in Figure 4.3.

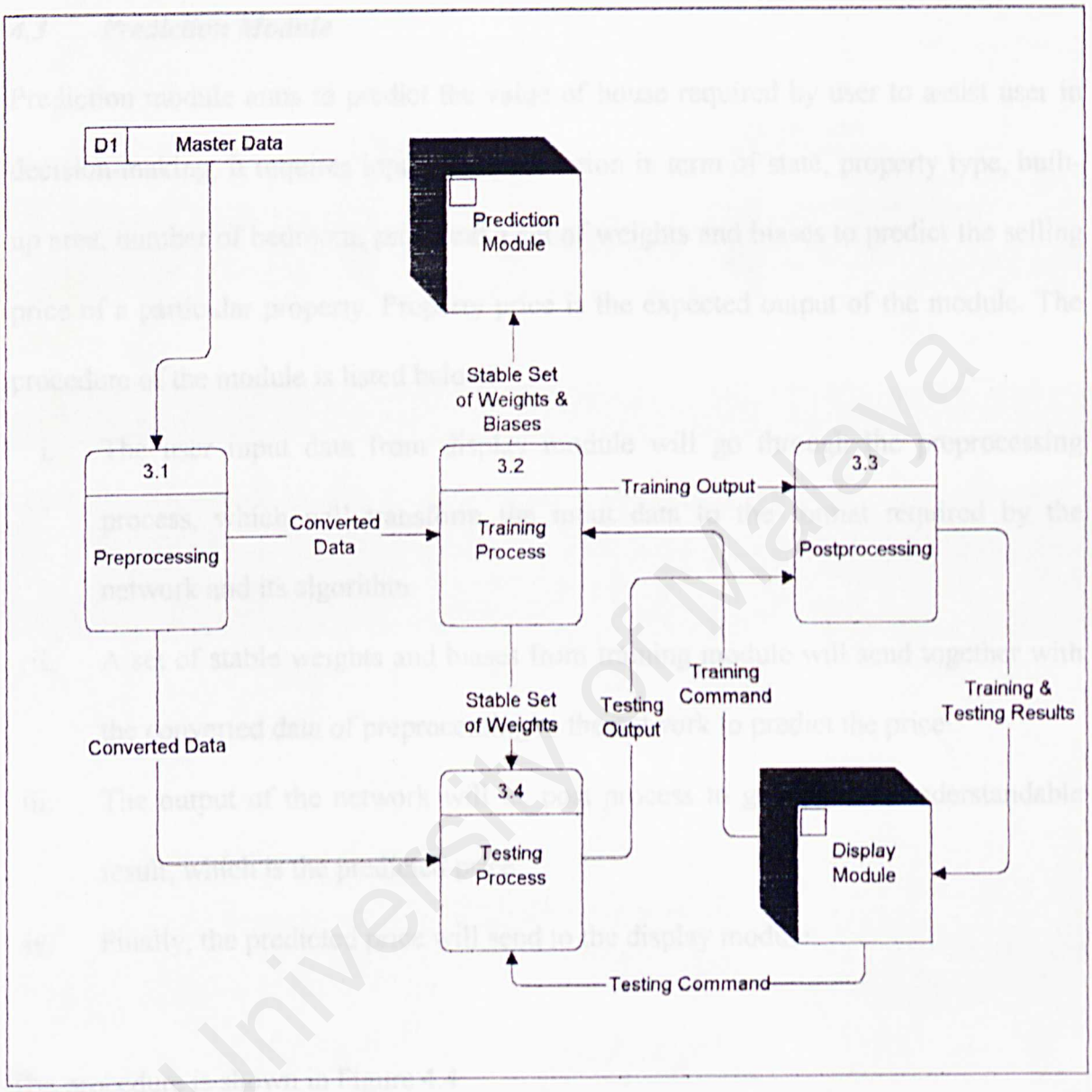

Figure 4.3 Training module data flow diagram

4.3 *Prediction Module*

Prediction module aims to predict the value of house required by user to assist user in decision-making. It requires input data of location in term of state, property type, built-up area, number of bedroom, price and a set of weights and biases to predict the selling price of a particular property. Property price is the expected output of the module. The procedure of the module is listed below:

i. The user input data from display module will go through the preprocessing process, which will transform the input data to the format required by the network and its algorithm.

ii. A set of stable weights and biases from training module will send together with the converted data of preprocessing to the network to predict the price.

iii. The output of the network will be post process to get the user understandable result, which is the predicted price.

iv. Finally, the predicted price will send to the display module.

The procedure is shown in Figure 4.4.

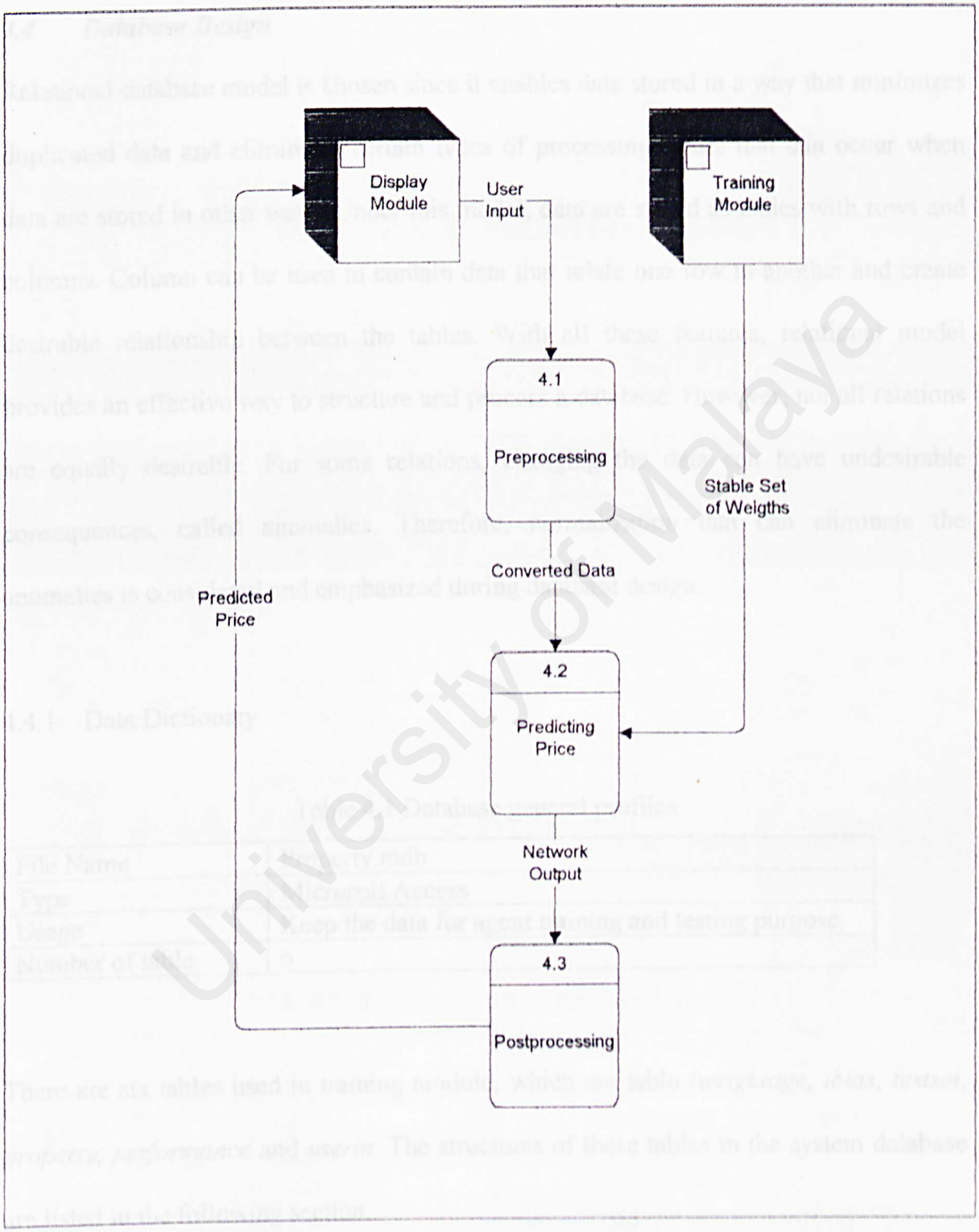

Figure 4.4 Prediction module data flow diagram

4.4 *Database Design*

Relational database model is chosen since it enables data stored in a way that minimizes duplicated data and eliminates certain types of processing errors that can occur when data are stored in other ways. Under this model, data are stored as tables with rows and columns. Column can be used to contain data that relate one row to another and create desirable relationship between the tables. With all these features, relational model provides an effective way to structure and process a database. However, not all relations are equally desirable. For some relations, changing the data can have undesirable consequences, called anomalies. Therefore, normalization that can eliminate the anomalies is considered and emphasized during database design.

4.4.1 Data Dictionary

Table 4.1 Database general profiles

File Name	Property.mdb
Type	Microsoft Access
Usage	Keep the data for agent training and testing purpose.
Number of table	6

There are six tables used in training module, which are table *iweightage*, *ibias*, *testset*, *property*, *performance* and *userin*. The structures of these tables in the system database are listed in the following section.

Table 4.2 Database structure for table *iweightage*

Field Name	*Data Type*	*Size*	*Description*
IW	Long integer	12	Input weight

Table 4.3 Database structure for table *ibias*

Field Name	*Data Type*	*Size*	*Description*
LW	Long integer	10	Layer weight
b1	Long integer	10	Input bias
b2	Long integer	10	Layer bias

Table 4.4 Database structure for table *testset*

Field Name	*Data Type*	*Size*	*Description*
iState	Long integer	2	Property state location code
iType	Long integer	2	Property type code
ibedrm	Long integer	2	Number of bedroom
iarea	Long integer	2	Property built-up area code
pprice	Long integer	10	Predicted price
Target	Long integer	10	Desired output
Error	Single	10	Differrence between pprice and target

Table 4.5 Database structure for table *property*

Field Name	*Data Type*	*Size*	*Description*
Price	Integer	10	Property price
StateCode	Integer	2	Property state location code
TypeCode	Integer	2	Property type code
Bedroom	Integer	2	Bedroom number code
AreaCode	Integer	2	Property built-up area code

Table 4.6 Database structure for table *performance*

Field Name	*Data Type*	*Size*	*Description*
accuracy	Integer	10	1 – mean squared error
id	Integer	1	Code to differentiate train and test performance

Table 4.7 Database structure for table *userin*

Field Name	*Data Type*	*Size*	*Description*
iState	Integer	2	Property state location code
iType	Integer	2	Property type code
ibedrm	Integer	2	Bedroom number code
iarea	Integer	2	Property built-up area code
pprice	Integer	10	Predicted price return by predictor

4.5 Screen design

Since the system is a Web based applications, its screen design is presented in the form of Web pages. To generate a better and user friendlier interface, the system's screen design are formatted in a standard layout so that various types of information, instructions, and messages always appear in the same general display area.

In general, the screen design is divide into tow parts, which are navigation bar and working area. Navigation bar is simply an index that guide visitors to find their ways to their interested pages directly, quickly and easily. Whereas, working area is an area that interacts with users' input and place to display the results. The standard layout effectively teaches users on how to use the system and getting users familiar with the system.

Besides, working area, which display information and messages always remain in a space that is wide and long enough for user to read the information. The screen design also provides command button, combo box, and default value for the text box to help and simplify the user input process. A sample of screen design was shown in Figure 4.5, which has its navigation bar in the left and working area in the right.

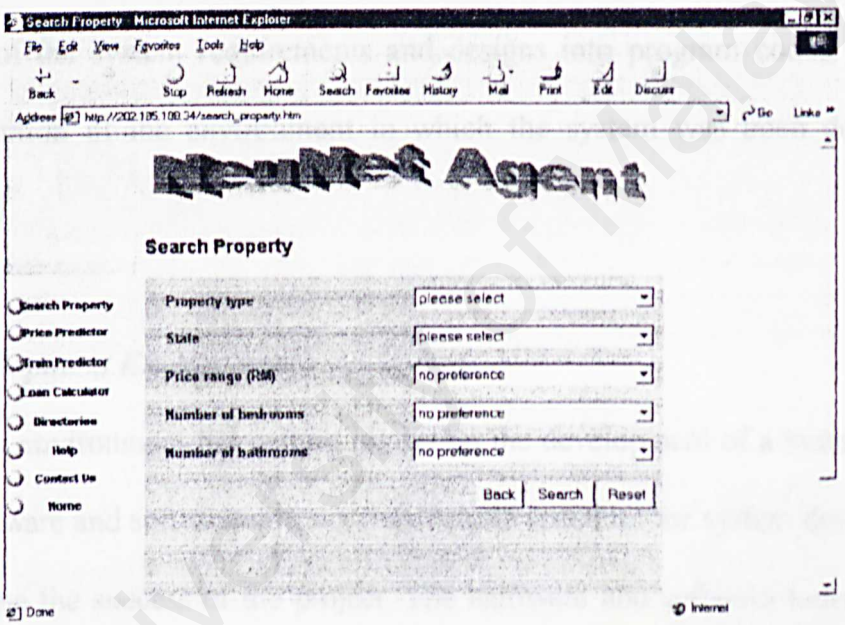

Figure 4.5 Search screen design

4.6 Summary

This chapter explains the conceptual and technical design of the system. It covers the system architecture design, where training module and prediction module was been emphasized, database design and screen design. These designs were lead to the implementation of the system that will be presented in the next chapter.

CHAPTER V

SYSTEM IMPLEMENTATION AND TESTING

System implementation is the construction of the system and the delivery of that system into "production". System implementation includes building and testing system, which is also called the construction phase. Construction phase of the system involves the conversion of the system requirements and designs into program codes. This chapter gives description of the environment in which the system was been developed and implemented.

5.1 Development Environment

Development environment has certain impact on the development of a system. Using the suitable hardware and software will not only help to speed up the system development but also determine the success of the project. The hardware and software tools that used to develop the entire system is as discuss below.

5.1.1 Hardware requirements

The hardware used to develop the system are listed below:

- 200MHz Processor
- 96MB RAM
- 2.1 GB Hard Disk

- Other standard desktop PC components

5.1.2 Platform Development

Before the system development could be carried out, the platform has to be first established. As the design of the system is a tier-to-tier architecture, the component used must be able to support each other. The setting up of the entire platform was difficult and time-consuming, which involves many components in the system.

5.1.3 System Users Environment

The first step in setting up the platform was installing the server operating system into the necessary computer. This process included formatting the disk to be NT file System format (NTFS). The NTSF format was chosen rather than normal FAT system because NTFS can provide a secured NT transaction across the platform. After that, the NT Server with Option Pack 4.0 was installed into the disk. During the installation a domain name was given to the server domain.

5.1.4 Web Accessible System

A web server is needed for entire project. Internet Information Server (IIS) was selected as the web server for the entire system. IIS 4.0 was installed through the installation of the Option Pack 4.0. Through the IIS manager, a virtual directory was created for the web access to the NeuNet Agent system.

5.1.5 Software Tools for NeuNet Agent Development

Table 5.1 below depicts the software used to develop the system.

Table 5.1 Summary of software used

Software	Purpose	Description
Microsoft Windows NT Server 4.0	System requirement	Operating system (OS)
Internet Information Server 4.0	System requirement	Web server host
Microsoft Visual InterDev 6.0	System development	Coding the web pages
Microsoft Visual Basic 6.0	System development	Coding the ActiveX components
Hyper Text Markup Language (HTML)	System development	Coding the web pages
Active Server Pages	System development	Coding the web pages
MATLAB 5.3	System development	Coding the Prediction Agent
Internet Explorer 5.0	System development	Viewing the web pages
Microsoft Access 2000	Database	Build the database and to store and manipulate the data

5.1.6 System with Database

The Microsoft Access Database was set up for keeping data of the system. Database storage was created for keeping all the NeuNet Agent data. In the created storage, the six tables as mentioned in section 4.4.1 were created for keeping the data used in training and prediction modules of NeuNet Agent.

In order to map the database to the web server, a system Data Source Name (DSN) was created for the system database storage. In this case, a DSN was registered through the ODBC in the web server machine. The ODBC could map to the storage by just calling the DSN.

5.2 Development of Webs

5.2.1 Microsoft Visual InterDev 6.0

At the initial stage of development phase, developer is required to create a web project for the system using Microsoft Visual InterDev 6.0. NeuNet Agent appropriately use some of the Microsoft Visual InterDev 6.0 features and technologies in creating, editing, deploying and managing its web site. Microsoft Visual InterDev 6.0 combines a rich set of database connectivity tools, wizards and design-time controls to increase the functionality and decrease the developemt time to build Active Server Applications. Some of the features of Microsoft Visual InterDev 6.0 are listed below:

i. RAD Environment

The new IDE provides a complete set of rapid application development (RAD) tools to let professional developer design, build, debug and deploy data-driven web

application faster than before. These include source code preserving WYSIWYG page editor with full support for dynamic HTML, complete and end-to-end debugging tools for client and server side codes as well as site design and management tools.

ii. Integrated Database Tools

Microsoft Visual InterDev 6.0 provides a complete set of tools for integrating databases with dynamic web application. Database features include drug and drop binding of database to HTML forms and reports, database design tools for creating and modifying database.

iii. Improved Web Application Programming Model

Microsoft Visual InterDev 6.0 simplifies the inherent complexities of building web application by providing an intuitive programming model, which includes: object-based and event driven programming, simple consistent programming model for both broad-reach and dynamic HTML-based application.

5.2.2 Web-page Development

Most of web pages in NeuNet Agent are dynamically generated, where page information is gathered from database. However, there are still some static pages for information display purposes. All web pages are coded into ASP document before being presented to the browser. ASP uses VBScript to convert coded processing logic into native HTML. The pure HTML output is accessible by any HTML-compatible browser.

VBScript, which is the foundation of ASP, is used for server-side execution in the system. Active Server Scripts are distinguished from HTML tags and normal content by using <% and %> delimiters. For client-side scripting, they must be delimited by <SCRIPT> and </SCRIPT>. For example in NeuNet Agent, VBScript was used to pre-check submitted form before transmitting them to the server. This could reduces the connection overhead, as well as making field validation immediate instead of having to wait for the server to parse the form, validate the data and generate a response documents.

System development involves endless cycle of coding, testing and modifying the source code. Testing was done by previewing in browser using Microsoft Visual InterDev 6.0 or opening the particular ASP page using Microsoft Internet Explorer.

5.3 Development of Prediction Agent

5.3.1 MATLAB 5.3

Prediction Agent was been built by using MATLAB and companion toolboxes, which provide an environment for technical computing applications. The following are some specific features that caused MATLAB to be chosen.

i. M-files

Unlike most traditional languages, MATLAB gives the freedom to focus on high-level technical concepts and ignore such programming details as memory management and variable declarations. M-files require no compiling or linking, it can be edits and debug and test for changes immediately, without leaving MATLAB [11].

M-files let user capture command-line explorations as permanent, reusable MATLAB functions. Standard flow-control constructs, such as if-else-else if and while loops, help in writing easy-to-read, well-designed code. M-file programs can incorporate any MATLAB feature, including user-definable structures or classes and a range of data types, from scalars and matrices to multidimensional arrays.

ii. MATLAB Toolboxes

Toolboxes are collections of algorithms that provide application-specific numerical, analysis, and graphical capabilities. With these algorithms, a number of techniques can be compare and apply without writing code.

MATLAB toolboxes provide application-specific functions, GUIs, and custom plot types for tasks requiring signal and image processing, data analysis and modeling, mathematics, finance and control system design. Neural Network Toolbox provides tools for the design, implementation, visualization, and simulation of neural networks [11]. In this system, the neurocontrol applications within the Neural Network Toolbox were been used to develop the Prediction Agent. Meanwhile, the Database Toolbox was been used to communicate with the ODBC/JDBC compliant database, which in this case the Microsoft Access.

5.3.2 Neural Network Development

Prediction Agent is a back propagation neural network with supervised network architecture that is supported by the Neural Network Toolbox. Besides, this toolbox provides a comprehensive set of training and learning functions and a modular network representation that allow an unlimited number of input set layers and network interconnections.

i. Neural Network Architecture

To develop a neural network, the function *newff* was been used to create a trainable feed-forward network. In order to develop an intelligent prediction agent, the total number of 18 neurons was been selected for the input layer among the choices of 10, 15 and 20 neurons due to its consistency and the faster speed in reaching the performance goal during the network training. The samples of training process that had been tried out during the selection are presented in Appendix A. One neuron was set for hidden layer to produce one output vector that is the predicted price. The function *sim* was then been used to simulate the network. It takes the network input and network object as its parameters and returns the network output [12].

ii. Neural Network Training

The network was trained in incremental mode using function *adapt* that takes the network object and the targets from the training set, and returns the trained network object, outputs and errors of the network for the final weights and biases. The fastest training algorithm for moderate size, Levenberg-Marquardt

algorithm provided by function *trainlm* was been chosen for the network. This is due to its fast training speed and memory reduction feature for use when the training set is large [12].

5.4 *Development of ActiveX Component*

To create a web-based Prediction Agent, an ActiveX component was needed to invoke the Prediction Agent that was written in the script of MATLAB M-file. Visual Basic 6 was been used to create the ActiveX component named *ProjAgt.dll*. This component was compiled as DLLs to run as in-process, non-visual application servers that will perform specific tasks. These tasks include invoke the MATLAB application in the server, run the prediction program, train and test the agent.

In order for other COM-based programs to user the DLL, the component DLL was been registered with the machine that will call the DLL. This process was done in two steps, which are registered with Regsrv32 utility followed by adding the registered component into the transaction package of the Microsoft Transaction Server (MTS) named ProjRunML.

Once the component has been created and registered, the Active Server scripts had been modified to use the component's methods. This will integrate the ActiveX component into the ASP. To call the component from the ASP, the *Server.CreateObject* method was used to create the object [13]. Then the object's methods that serve the tasks mentioned as above will invoke by the code *ProjAgt.**methodname***.

5.5 *System Testing*

Testing is the process of exercising or evaluating a system by manual or automatic means to verify that it satisfied requirements or to identify differences expected and actual results. Testing is probably the least understood part of a software development project. A bug is any unexpected, questionable, or undesired aspect or behavior displayed, facilitated, or caused by the software being tested. Testing can uncover different classes of errors in a minimum amount of time and with a minimum amount of effort. The strategies used for testing are unit testing, integration testing and system testing. Nevertheless, the Prediction Agent was been tested for its intelligence.

5.5.1 Unit Testing

Unit testing was done in a controlled environment whenever possible, as to feed a predetermined set of data to component being tested and observed what output actions and data produced. In addition, the internal data structures, logic and boundary conditions for input and output data was been checked. The following sections explain those testing strategies that were carried out throughout the development of the system.

i. Review/examining the code

One of the basic and important testing strategy is code reviewing. During reviewing the code, the correctness of coding was to be review and identified by always comparing it to the original design of the program flow. When the logic and flow of the program were identified, the code was commented so that it can be traced in the future.

The code was also examined and debugged in order to identify any fault coding. Coding with ASP was difficult, as there were no proper ASP debugger and tester used in the project. The examination and debugging of ASP code was worked out by adding a "watch" line in the code.

ii. Testing with test cases

There is no enough if only reviewing the coded. There should be a more practical strategy in order to identify the variance between the prototype and the requirement. Therefore, the page has to be tested with some test cases. The test cases used in the project were a set of structural input for web page and sets of property data for the network. This will allow the reaction of the page to the input data to be tested. This could identify the page's faults, which probably happen in normal condition.

5.5.2 Integration Testing

When the individual components are working correctly and meet the objectives, these components are combined into a working system. In other words, integration testing is the process of verifying that the system components work together as described in the system and program design specifications.

During the integration, all the module prototypes were combined and tested in a testing environment. The testing environment was consistent for all modules in terms of interface and function calling procedures. To do this, the program flow and testing needs for each of the modules were reviewed and identified. After identifying the testing requirements for

the integration, the program flow of the entire system were reviewed and tested. When all the modules were integrated, the entire system was tested with some test cases.

5.5.3 System Testing

The last testing procedure done is system testing. It is a series of test being carried out to fully exercise parallel in the system, which aims to ensure that the system does what the users want it to do.

The system testing was done by first browsing the web application, tried all of the available tools and services. Then all modules were tested for its performance efficiency in performance testing. Meanwhile, data integrity testing was carried out to verify that the data is stored in a manner where it is not compromised under updating, restoration or retrieval processing.

5.5.4 Testing For Intelligence of Prediction Agent

The agent was been tested for its intelligence by using the test data set. The data set that was used is new data for the agent. These data was presented to the agent without showing its target data. Target data that is the actual property price, is used to compare with the agent predicted output to calculate the mean squared error. The calculated value was used to verify the achievement of the agent. If the value is less than the performance goal of 1×10^{-6}, then the agent is said to be intelligence enough in handling its task.

5.6 *Summary*

The implementation of the system was been covered in terms of development environment, and developments of web pages, prediction agent and ActiveX component. Its also discussed how the system was been tested. System testing will then followed by the stage of system evaluation that comes on the next chapter.

CHAPTER VI

SYSTEM EVALUATION

Upon the completion of the project, the system strengths and limitations were evaluated by comparing with the requirement specification. This chapter will explains in detail about the evaluation of the system in terms of strength and limitations.

6.1 System Strengths

The following are the strengths of the system, which had achieve some of the proposed objectives.

6.1.1 Simple and User-friendly Interface

User interface in NeuNet Agent is easy to understand and user-friendly. Screen design is formatted in a standard layout where various types of information, tools, and messages always appear in the same general display area. The standard layout effectively teaches users on how to use the system and getting users familiar with the system. The screen design also provides command button, combo box with default value to help and simplify the user input process.

6.1.2 Validation on Input Data

The system was developed to be robust enough to handle any invalid input into the system. Client side scripting will check for input validation before the form is submitted and generate appropriate feedback to user whenever an invalid input is encountered.

6.1.3 Intelligent Prediction Tool

Prediction Agent is a price prediction tool with embed neural network, which is intelligent enough to make prediction of house price based on state location, property type, number of bedroom and range of built-up area of the property. This module was designed with emphasis on its supportive role as a tool for assisting user in price comparison and decision-making.

6.1.4 Reliable Prediction Tool

Prediction Agent is a reliable tool as it can be trained for better performance. Mean squared error will be calculated and shown to user as the testing result. This result represents the accuracy of prediction and as a guideline for user to determine the reliability of agent.

6.1.5 Easy Accessibility

This system is a web-based application, which can be accessed easily using the Internet Explorer web browser. The services provided are available to all users without requesting for member login.

6.2 *System Limitations*

Due to project boundaries, there are some limitations in NeuNet Agent. The limitations are stated as below:

6.2.1 Browser Limitation

NeuNet Agent can only run in Internet Explorer 4.0 and above. NeuNet Agent requires a browser that can interpret VBScript, the default supporting language for ASP. Browsers that do not support these features will not be able to perform the available functions in the system.

6.2.2 Prediction Limitation

Due to the time constraints, the Prediction Agent does not reflect a real predictor. Prediction was made based on four criteria, which are location in term of state, house type, number of bedrooms and built-up area of the property. In real case, prediction made should also consider other factors such as age of the property, location in term of city or town, number of bathrooms, public facilities available nearby and so on.

6.2.3 Security Limitation

Since only administrator is allow to train the Prediction Agent, the permission to access the training page was been control at the file level. The system will prompt for the authorization when the train option was invoked. However, the username and password are being passed in non-encrypted form from the Web browser to the Web server [13].

This will enable a potential hacker to intercept packet and retrieve the NT username and password.

6.2.4 On-line Help File

On-line help file is very important in any applications. Due to time constraints, NeuNet Agent does not include the on-line help file but a complete user manual is included as in Appendix II for user reference.

6.3 Summary

The strength of the system was presented together with the system limitations in this chapter as the output of system evaluation. This is lead to the conclusion that will be presented in Chapter VII.

CHAPTER VII

CONCLUSION

The problems encountered during the development of the system will be covered in this chapter. Besides that, recommendations for future enhancements of the system were also included. This chapter ends with conclusion of the project.

7.1 Problems Encountered and Solutions Taken

A lot of system analysis need to be done on technologies and programming concepts and AI approaches before starting to develop NeuNet Agent. The basic knowledge needed as a foundation in building an application of this nature involves studies in fields such as the Internet, intelligent agent, e-commerce and neural network application. The following are some of the major problems encountered during the process of system development.

7.1.1 Difficulties in Choosing Development Technology, Programming Language and Tools

There are many software tools available to develop a web-based database system currently. Choosing a suitable technology and tool was a critical process as all tools have their strengths and weaknesses. In addition, the availability of the required tool for

development was a major consideration. A tough decision was needed to choose from Active Server Pages technology, CGI or Java.

In order to solve this problem, seeking advised and views from project supervisor and course mates were carried out. Surfing the Internet and referring to reading materials had helped in clarify some doubts.

7.1.2 Difficulty in Setting Up the Server and Platform

It is time consuming due to new exposure and lack of experience in setting up the server and platform. The major problem occurs while configuring the server during setup. Any incorrect configuration will cause improper setup of the server and lead to problems that occur during installation of development tools. Hence, more time and effort had been spent to setup the server and working platform.

Getting advises and guidance from experienced senior and discussion with course mates had helped to overcome this problem. Besides, reviewing the relevant documents also help in solving the difficulty.

7.1.3 Inexperience in the Chosen Programming Language

The new exposure of programming language such as ASP and MATLAB had increased the learning curve before starting the development of NeuNet Agent.

The problem was solved through surfing the Internet for related materials and referring to the reference books available in the market. Discussion with friends using the same technology was a great help. A more efficient solution was through trial and error during the coding phase.

7.1.4 Difficulty in Building Neural Network Agent

Neural network approach was used to build the prediction agent. There are many steps involved in the developing neural network application, which include data collection for training and testing the network, defining network architecture and learning method, initialization of the network weights and parameters, transformation of data to network inputs and, training and testing the network. These steps were time-consuming because each step involved the process of testing, modification and feedback revising. Lack of knowledge in neural network and inexperienced in using MATLAB had lengthens the time spent in developing the agent.

Reference books and on-line resources had helped in the understanding of neural network architectures. Help link and the user manual provided by MATLAB had become the main source in developing the network. Trial and error approach was been carried-out to determine the most suitable network parameters and weights for the agent. Internet surfing also had helped in data set collection.

7.2 Future Enhancements

Future enhancement can be done to make the system more intelligent and ease to use. A system development knows no boundaries as new requirements and better implementation methods continue to arise and evolve. The following are several enhancements that could extend the usability of the developed system.

7.2.1 Browser Independent

In future, NeuNet Agent should not only limited to Microsoft Internet Explorer but also able to support various types of browser.

7.2.2 More Effective Training Data Set

Training the agent involved presenting the training data set to the Agent so that the weights can be adjusted to produce the desired output for each of the inputs. In the ideal case, the agent can learn the features of the input data. Thus, with the presentation of novel inputs that are not identical to those in the training set, the Agent would become more intelligent in making prediction.

7.2.3 Enhance Administration Task

Administration task can be further enhanced to include more features to ease agent maintenance process. These include analytical tool for viewing agent performance, data set upgrade function and data set backup.

7.2.4 Security Enhancement

The username and password that is required for accessing the agent training is currently transmitted to the server in non-encrypted form. To prevent the interception of hacker, the username and password should be transmitted over a Secure Socket Layer or other encrypted channel [13].

7.2.5 On-line Help File and Documentation Help System

On-line help file can be incorporate into NeuNet Agent. Demonstration basis help system may also be incorporated to reduce the system learning curve to enhance usability among its users.

7.2.6 Attractive User Interface

NeuNet Agent will become better publicized if its homepage is enhanced to be more attractive and interactive by adding more meaning and user-friendly images, property location maps, animation images and sounds. Showing the performance in the form of graph will ease the user in analyzing Agent response in training and testing process.

7.3 Conclusion

The project has met its objective of developing an agent that is intelligent enough to assist user in the activities of electronic commerce, as were defined during the analysis stage and fulfills all the functional and non-functional requirements of the system. NeuNet Agent is a fully operational web-based intelligent e-commerce tool, which was found to be user-friendly, easily understood, reliable and intelligent.

However, NeuNet Agent still has its limitations, where it is only limited to the scope of Malaysia residential property and unable to provide a prediction tool that reflect a real predictor. These major limitations are hard to tailor at the moment due to some constraints. In future, the scope of implementation of NeuNet Agent may be extended to included wider region and the increase the accuracy of the Prediction Agent by consider more factors that give effects for the property price, and thus will be able to provide a comprehensive web-based intelligent tool for e-commerce industry.

Despite those limitations, the project has reasonably achieved all its objectives. Most importantly, the project has brought a step closer to the understanding of neural network application in e-commerce. Throughout the development process, valuable knowledge was gained from the complexities and intricacies of the programs. Among them are concepts of intelligent agent, e-commerce, web technology as well as the configuring a web server and working with the Network operating system. Programming in MATLAB, ASP, HTML, VBScript prove to be valuable experience. Theories and

knowledge gained throughout the course of computer science studies like System Analysis and Design, Software Engineering and Artificial Neural Network were literally put into practice.

7.4 Summary

Problems encountered during the system development were presented together with the solution taken. Suggestions for future enhancement were made. Finally this chapter ends with the conclusion of the overall project development.

BIBLIOGRAPHY

[1] Mohamed H. Hassoun, *Fundamental of Artificial Neural Network*, Bradford Book, 1995.

[2] J.G. Taylor, *The Promise of Neural Network*, Springer-Verlag, 1993.

[3] Phillip D. Wasserman, *Advanced Methods in Neural Computing*, Van Nostrand Reinhold, 1993.

[4] Financial Predictions with Neural Networks, http://www.calsci.com/tsFinance.html Date referred: 1 July 2000.

[5] Using an Intelligent Agent to Enhance Search Engine Performance, http://www.firstmonday.dk/issues/issue2_3/jansen/index.html Date referred: 17 June 2000.

[6] Electronic Commerce, http://www.ecentre.org.uk/frameset_business.asp Date referred: 19 June 2000.

[7] Ian S. Graham, *HTML 4.0 Sourcebook*, John Wiley & Sons, Inc., 1998

[8] John Paul Ashenfelter, *Choosing a Database for Your Web Site*, John Wiley & Sons, Inc., 1998.

[9] Craig Eddy and B. Haasch, *Web Programming with Visual Basic*, Sams Publishing, 1996.

[10] Introduction to the Matlab Neural Network Toolbox, http://carol.wins.uva.nl/~portegie/matlab/nnt/ Date referred: 27 July 2000.

[11] Neural Network Product Description, http://www.mathworks.com/products/neuralnet/ Date referred: 18 November 2000.

[12] Howard Demuth and Mark Beale, *Neural Network Toolbox For Use With MATLAB*, The Math Works, Inc., 1998.

[13] Scot Johnson, Keith Ballinger and Davis Chapman, et al., *Special Edition Using Active Server Pages* [TM], Que Corporation, 1997.

APPENDICES

APPENDIX A

SAMPLES OF TRAINING PROCESS

The following are the samples of training process done with different number of neurons used for input layer. This samples was used to determine the number of neurons in the hidden layer, where in this case, 18 neurons was been selected as the mean of 15 and 20 neurons.

a. Sample of 10 neurons

Sample 1:

» neuron10

TRAINLM, Epoch 0/300, MSE 3.67103/1e-006, Gradient 756.915/1e-010
TRAINLM, Epoch 50/300, MSE 2.2439e-006/1e-006, Gradient 3.98962e-005/1e-010
TRAINLM, Epoch 100/300, MSE 2.21628e-006/1e-006, Gradient 4.88346e-005/1e-010
TRAINLM, Epoch 150/300, MSE 2.20126e-006/1e-006, Gradient 7.72558e-005/1e-010
TRAINLM, Epoch 200/300, MSE 2.14252e-006/1e-006, Gradient 9.78994e-005/1e-010
TRAINLM, Epoch 250/300, MSE 1.71541e-006/1e-006, Gradient 0.000300762/1e-010
TRAINLM, Epoch 300/300, MSE 1.16059e-006/1e-006, Gradient 0.00726736/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.

b. Samples of 15 neurons

Sample 1:

» neuron15

TRAINLM, Epoch 0/300, MSE 6.42916/1e-006, Gradient 1002.89/1e-010
TRAINLM, Epoch 50/300, MSE 1.73361e-006/1e-006, Gradient 0.000199287/1e-010
TRAINLM, Epoch 100/300, MSE 1.72585e-006/1e-006, Gradient 0.00251823/1e-010
TRAINLM, Epoch 150/300, MSE 1.59357e-006/1e-006, Gradient 6.2647e-005/1e-010
TRAINLM, Epoch 200/300, MSE 1.59142e-006/1e-006, Gradient 0.00123709/1e-010
TRAINLM, Epoch 250/300, MSE 1.58796e-006/1e-006, Gradient 0.00137734/1e-010
TRAINLM, Epoch 300/300, MSE 1.57255e-006/1e-006, Gradient 0.000147282/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.

Sample 2:

» neuron15

TRAINLM, Epoch 0/300, MSE 3.839/1e-006, Gradient 749.221/1e-010
TRAINLM, Epoch 50/300, MSE 1.30032e-006/1e-006, Gradient 7.67526e-005/1e-010
TRAINLM, Epoch 100/300, MSE 1.29666e-006/1e-006, Gradient 0.000385836/1e-010
TRAINLM, Epoch 150/300, MSE 1.29143e-006/1e-006, Gradient 0.000117947/1e-010
TRAINLM, Epoch 200/300, MSE 1.28652e-006/1e-006, Gradient 3.6955e-005/1e-010
TRAINLM, Epoch 250/300, MSE 1.2817e-006/1e-006, Gradient 5.46088e-006/1e-010
TRAINLM, Epoch 300/300, MSE 1.27716e-006/1e-006, Gradient 9.19057e-005/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.

c. Samples of 18 neurons

Sample 1:

» neuron18

TRAINLM, Epoch 0/300, MSE 1.62732/1e-006, Gradient 219.305/1e-010
TRAINLM, Epoch 50/300, MSE 1.83951e-006/1e-006, Gradient 1.28647e-005/1e-010
TRAINLM, Epoch 100/300, MSE 1.83512e-006/1e-006, Gradient 5.14084e-006/1e-010
TRAINLM, Epoch 150/300, MSE 1.75919e-006/1e-006, Gradient 7.94308e-005/1e-010
TRAINLM, Epoch 193/300, MSE 9.65273e-007/1e-006, Gradient 0.00612787/1e-010
TRAINLM, Performance goal met.

Sample 2:

» neuron20

TRAINLM, Epoch 0/300, MSE 0.362/1e-006, Gradient 139.626/1e-010
TRAINLM, Epoch 50/300, MSE 1.35276e-006/1e-006, Gradient 0.000477229/1e-010
TRAINLM, Epoch 100/300, MSE 1.31074e-006/1e-006, Gradient 0.00104527/1e-010
TRAINLM, Epoch 150/300, MSE 1.23257e-006/1e-006, Gradient 0.000135707/1e-010
TRAINLM, Epoch 184/300, MSE 9.87983e-007/1e-006, Gradient 0.0123023/1e-010
TRAINLM, Performance goal met.

Sample 3:

» neuron20

TRAINLM, Epoch 0/300, MSE 8.16521/1e-006, Gradient 1256.37/1e-010
TRAINLM, Epoch 50/300, MSE 2.92588e-006/1e-006, Gradient 0.00177162/1e-010
TRAINLM, Epoch 100/300, MSE 2.71725e-006/1e-006, Gradient 0.000783241/1e-010
TRAINLM, Epoch 150/300, MSE 2.68768e-006/1e-006, Gradient 0.000314544/1e-010
TRAINLM, Epoch 200/300, MSE 2.66577e-006/1e-006, Gradient 0.00102854/1e-010
TRAINLM, Epoch 250/300, MSE 2.62262e-006/1e-006, Gradient 0.000911571/1e-010
TRAINLM, Epoch 300/300, MSE 2.58127e-006/1e-006, Gradient 0.00251638/1e-010
TRAINLM, Maximum epoch reached, performance goal was not met.

APPENDIX B

FIGURES OF TRAINING PROCCESS

The following are the figure generated during the network training process.

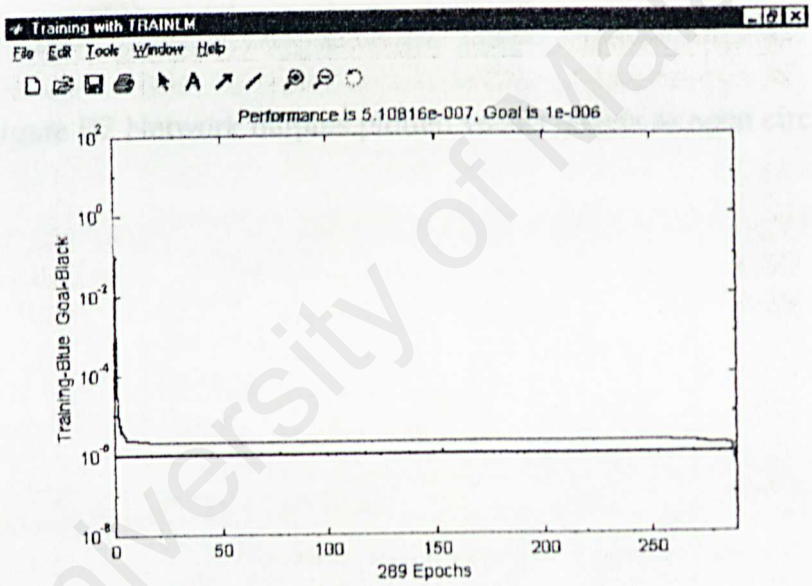

Figure B1 Graphical output of a sample of training result

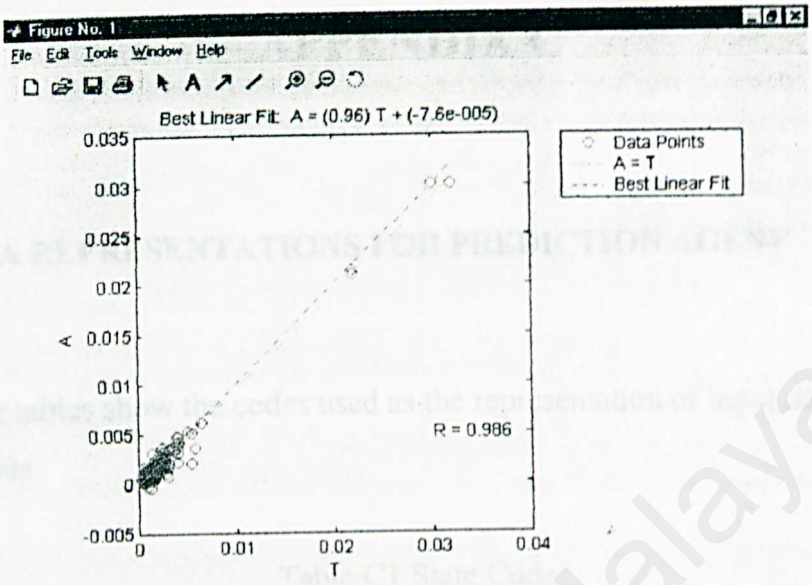

Figure B2 Network outputs plotted versus targets as open circle

APPENDIX C

INPUT DATA REPRESENTATIONS FOR PREDICTION AGENT

The following tables show the codes used as the representation of input data for the prediction agent.

Table C1 State Code

State Name	State Code
Perlis	1
Kedah	2
Kelantan	3
Terengganu	4
Sabah	5
Sarawak	6
Pahang	7
Perak	8
Malacca	9
Negeri Sembilan	10
Penang	11
Selangor	12
Johore	13
Kuala Lumpur	14

Table C2 Type Code

Type Name	Type Code
Flat	1
Town House	2
Apartment	3
Terraced – Single Storey	4
Terraced – Double Storey	5
Condominium	6
Semi-detached House	7
Bungalow	8

Table C3 Bedroom Code

Bedroom Number	Bedroom Code
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10 and above	10

Table C4 Size/Area Code

Area Range (sq.ft.)	Area Code
below 799	1
800-1199	2
1200-1599	3
1600-1999	4
2000-2399	5
2400-2799	6
2800-3199	7
3200-3599	8
3600-3999	9
4000-4399	10
4400-4799	11
4800-5199	12
5200-5599	13
5600-5999	14
6000-6399	15
6400 and above	16

APPENDIX D

PREPROCESSING AND POST-PROCESSING

Preprocessing of input vectors for Prediction Agent:

- Input vector 1 = State Code / 10000
- Input vector 2 = Type Code / 10000
- Input vector 3 = Bedroom Code / 10000
- Input vector 4 = Area Code / 10000

Post-processing of predicted output vector:

- Predicted Price = Output vector x 10^{8}

APPENDIX E

ARCHITECTURE OF PREDICTION AGENT

Network Type: Back propagation feed forward neural network

Training function: Levenberg Marquardt Back propagation

Number of input vector: 4

Number of output vector: 1

Table E1 Network architecture

Layer	Hidden Layer	Output Layer
Number of neuron	18	1
Transfer function	Tan-sigmoid	Bipolar linear

APPENDIX F

USER MANUAL

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

NeuNet Agent is an Internet based application. It is designed to collects relevant residential property information from the web to assist users in decision-making based on the domain of house available for sale in Malaysia. Besides that, it provides an intelligent tool for price prediction. It is easy to use and learn. All the functions in this system can easily be executed by a simple point and click on the available function button.

The rationale of this system is to build an intelligent agent for the following purposes.

i. To simplify the property searching process on the web.

ii. To collect relevant information on a particular house specified by user.

iii. To enable price comparison with predicted price.

iv. To provide other useful online tools related to house purchasing.

This user manual will guide you through all the functions available in the system. This manual includes the following parts:

- System Overview and Essentials
- User Section User Manual
- Administration Section User Manual

CHAPTER 2: HARDWARE AND SOFTWARE REQUIREMENTS

2.1 Hardware Requirements

The listing below shows the minimum hardware requirements that are needed to access the NeuNet Agent system.

i. A 496 processor or above (Pentium processor is recommended)

ii. A16MB of RAM or greater

iii. A 1.2GB of Hard Disk or greater

iv. Modem/ NIC to connect to the Internet/ Web Server

v. A SVGS Graphic Adapter

vi. Keyboard and Mouse as input devices

2.2 Software Requirements

The listing below shows the software requirements that are needed to run the NeuNet Agent system for server and client machine respectively.

Server Machine

i. Windows NT 4.0 with Internet Information Server 4.0

ii. Microsoft Internet Explorer 4.0 or above

iii. Microsoft Access

iv. MATLAB version 5.3

Client Machine

i. Windows 95 or Windows 98 or Windows NT

ii. Microsoft Internet Explorer 4.0 or above

CHAPTER 3: GETTING STARTED

NeuNet Agent is a web application that can be accessed through typing the address of the site in the web browser. Before accessing NeuNet Agent web site, make sure that your computer meets the minimum hardware and software requirements as stated in the previous chapter.

3.1 From Microsoft Internet Explorer

To access NeuNet Agent, the Microsoft Internet Explorer web browser needs to be start. Then in the address test box area, type the URL http://anx307/default.htm or IP address http://202.185.108.34/default.htm of NeuNet Agent and press the enter to connect to the server. A successful connection will cause the web browser to display the main page of the NeuNet Agent as shown in figure 3.1 below.

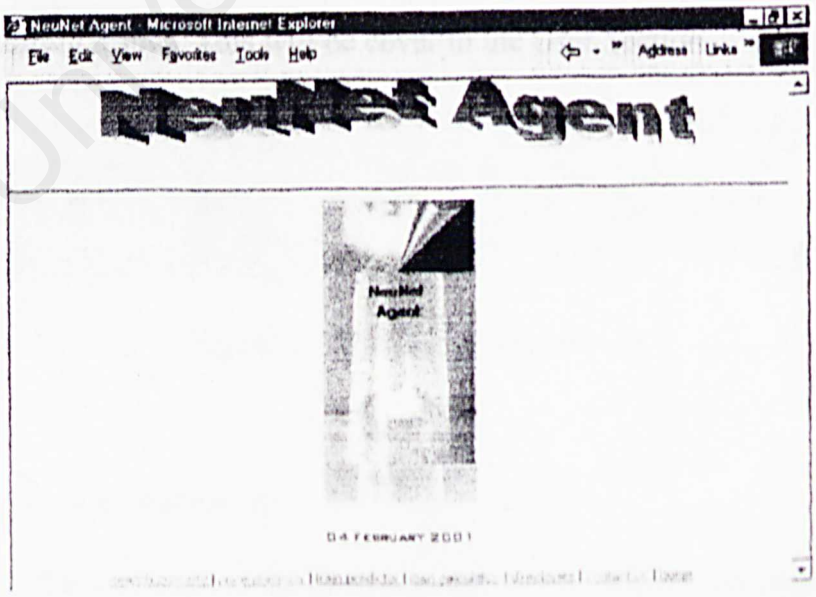

Figure 3.1 NeuNet Agent main page

Within this page you can enter the various links provided as shown below:

- Search Property
- Price Prediction
- Train Predictor
- Loan Calculator
- Directories
- Contact Us
- Home

3.2 Important notes for using NeuNet Agent

User can access the all tools provided by the NeuNet Agent except for *Train Predictor*, which will train and test the Prediction Agent. Only valid administrator is allowed to access this page and train and test the Prediction Agent. However, user can view the training and testing results. This will be cover in the User Section.

CHAPTER 4: USER SECTION

There are three tools available for the user, which is Search Property, Price Prediction, Loan Calculator. These tools will be future describe in the following sections.

4.1 Search Property

This tool enables the user to search for specific property required by user. User needs to select all the fields provided in the search form before submit the form.

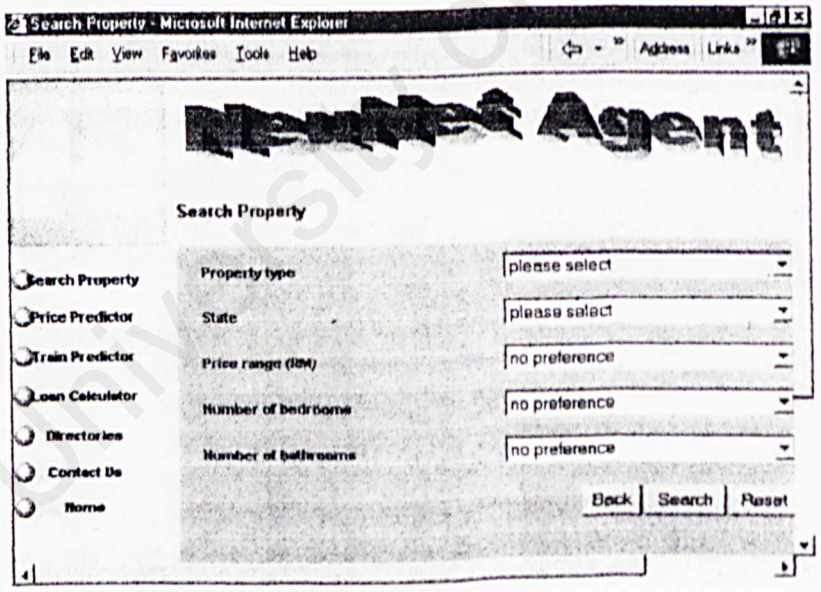

Figure 4.1 Search Property screen

The following are the functions of the command buttons:

- Back - Link the user back to the previous page before this current page.

- Search - Send the user selected items to the server and invoke the property searching process.
- Reset - Clear all the values selected by the user in the form and reset them to the initial state.

4.2 *Price Prediction*

The purpose of this tool is to make price prediction based on the property criteria input by the user. User needs to select a value for the four combo boxes provided. These combo boxes include property type, state location, number of bedroom and area range (sq.ft.).

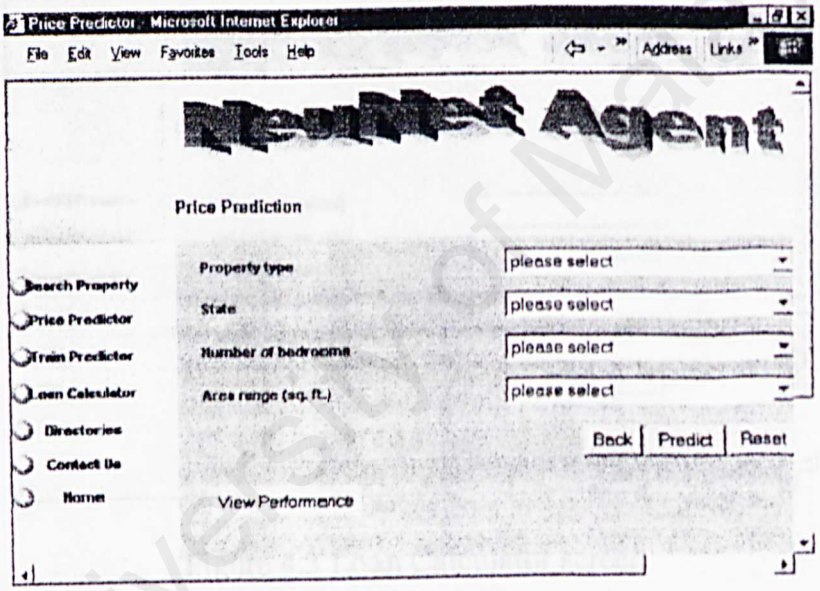

Figure 4.2 Price Prediction screen

Here are the functions of the command buttons:

- Back – Go back to the previous page.
- Predict – Make price prediction based on the input. The predicted output will be display to the user.
- Reset – Clear the selected value of combo boxes and set back to its initial value.
- View Performance – Display the performance detail of the Predictor.

4.3 *Loan Calculator*

This tool aims to provide the facility of loan calculation and return an amount of monthly installments as its output result to the user. The calculation works automatically once all the fields have been fill in.

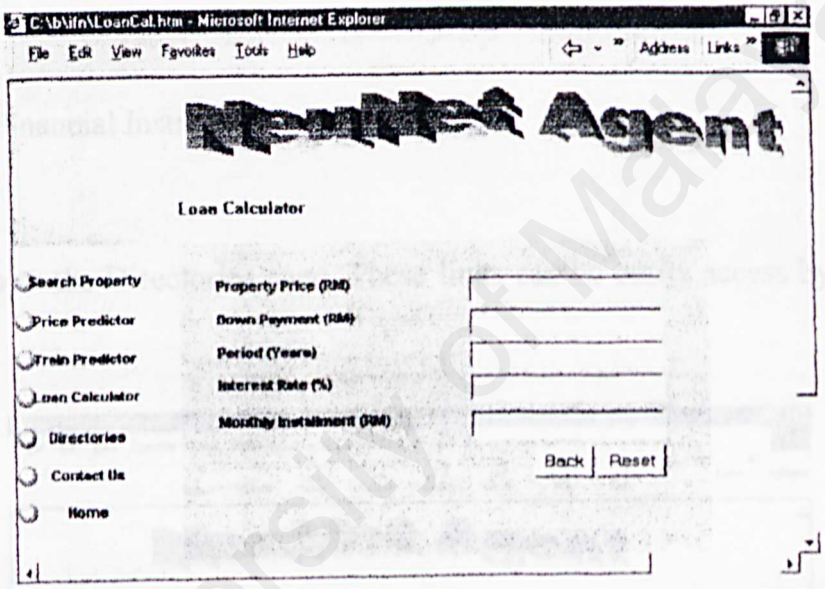

Figure 4.3 Loan Calculator screen

The functions of the command buttons are as below:

- Back – Go back to the previous page.
- Reset – Clear all the text fields.

4.4 *Directories*

This page contains hyperlinks to several resources that are related with the property field, which include the following:

- Insurance Companies
- Interior Designers
- Realtors
- Bank and Financial Institutions

Figure 4.4 shows the Directories page. These links can be easily access by clicking the hyperlink.

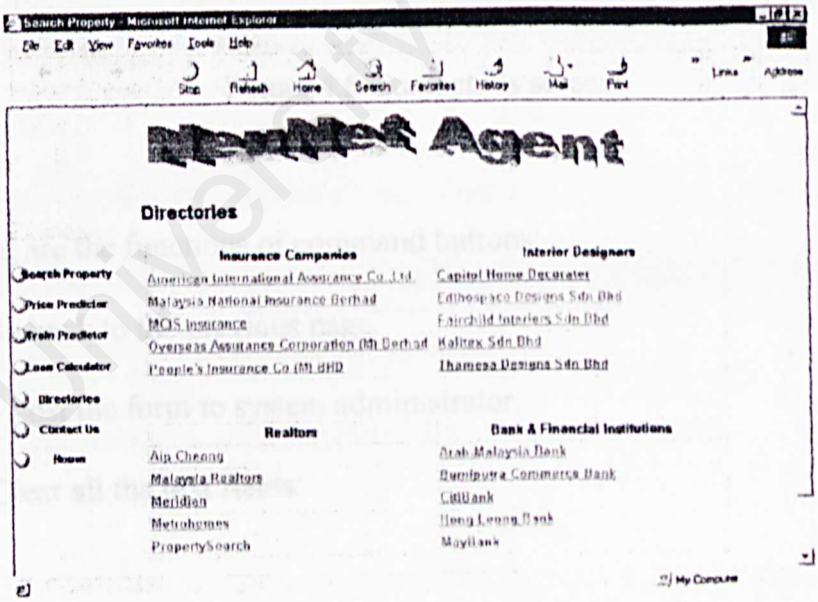

Figure 4.4 Directories screen

4.5 Contact Us

This page aims to enable user to contact with the system administrator. Once finish filling in the form, the comment or message will be post to administrator.

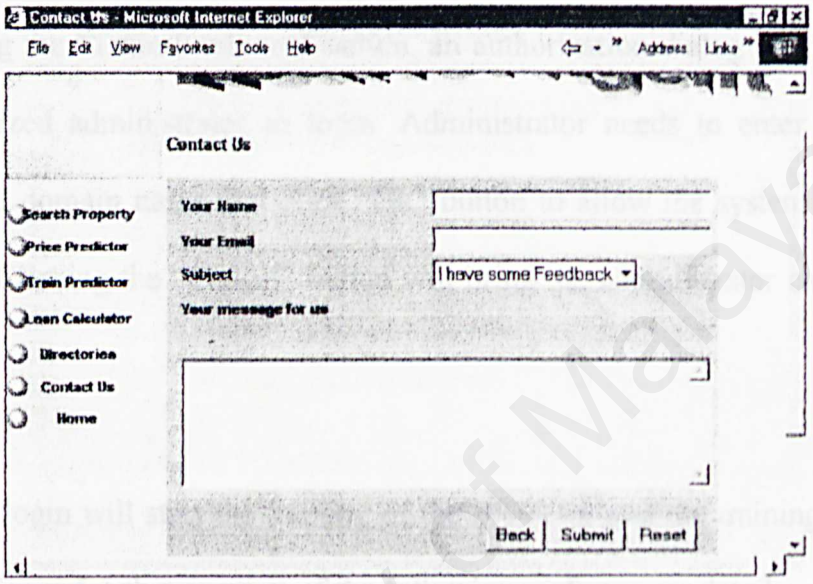

Figure 4.5 Contact Us screen

The following are the functions of command buttons:

- Back – Go back to the previous page.
- Submit – Post the form to system administrator.
- Reset – Clear all the text fields.

CHAPTER 5: ADMINISTRATOR SECTION

5.1 Train Predictor

After clicking the "Train Predictor" button, an authorization dialog box will prompt to allow authorized administrator to login. Administrator needs to enter the username, password and domain name and click "Ok" button to allow the system to process the verification. Clicking the "Cancel" button will bring the administrator back to previous page.

A successful login will start the training of the Predictor and the training result will be shown as in figure 5.1 that consists of the following performance details:

- Performance goal – the desired mean squared error.
- Training result – the successfulness of training.
- Mean squared error – the average squared error between the predicted price and the actual price.
- Prediction accuracy – the degree of accuracy in the testing.

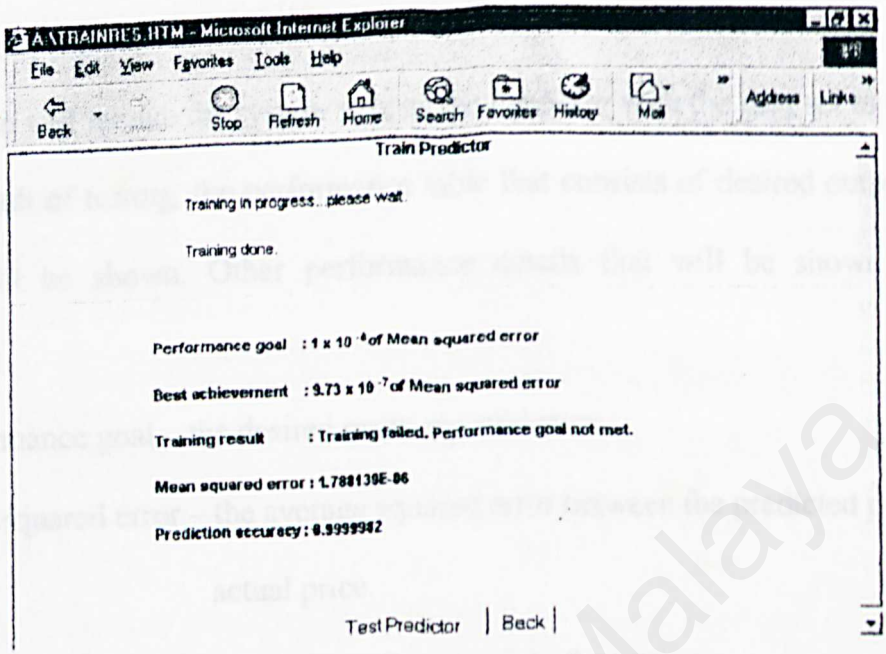

Figure 5.1 Result page of Train Predictor

This result page also provides two command buttons with functions as below:

- Test Predictor – Run to test the Predictor
- Back – Go back to the previous page

5.2 Test Predictor

This button will invoke the system to tests the Predictor with the data set in the database. As the result of testing, the performance table that consists of desired output and actual output will be shown. Other performance details that will be shown include the following:

- Performance goal – the desired mean squared error.
- Mean squared error – the average squared error between the predicted price and the actual price.
- Prediction accuracy – the degree of accuracy in the testing.