Priya

ACKNOWLEDGEMENT

The author would like to express her deepest gratitude to Dr. Selvanathan Narainasamy who has been a great guidance in helping the author prepare this proposal. The author would also like to thank Dr.Rodziati.

The author would like to take this opportunity to thank the authors family and friends without whose help this project would not have been able to be completed.

TABLE OF CONTENT

CHAPTER 4 SYSTEM IMPLEMENTATION

LIST OF FIGURES

ABSTRACT

Agent technology is becoming more popular among system designers who want their system to be more personalized, continuously running and semi- autonomous. These properties make agents useful for a wide variety of information and process management tasks. It should come as no surprise that these same qualities are particularly useful for the information-rich and process-rich environment of electronic commerce.

As such agent technology in electronic commerce (e-commerce) would be essential to improve on-line shopping which is becoming more and more popular these days. It is in these roles that agent system designers have the challenge to accurately identify and the opportunity to prescribe innovative ways of doing business.

This project would be to build an electronic house agent which uses agent technology. This agent mediated house agent would be able to search a few search engines simultaneously for information of the house which best meet the clients requirement. The information collected would be displayed on-line to the client.

This house agent would be able to change and improve the perspective of e-house buying in Malaysia. Buyers would be able to get the precise information on what they want. The buyer would not need to browse the net on his own. The agents help would be seeked to get the correct information wanted according to their requirement specification.

The agent would also be able to give its recommendation on which of the information displayed best suits the buyer.

CHAPTER 1 INTRODUCTION

INTRODUCTION

1.1 Project Definition

Shopping through Internet is becoming more and more popular in Malaysia. In the current trend of e-commerce, buying houses through the Internet has becoming a norm amongst Malaysian house buyers. Most buyers prefer to get a clear information on the house for sale before buying it. It is understood that even though there are many local web sites which advertises on houses for sale, be it of real-estate agencies or individual house owners who intend to sell their house, buyers find it difficult to get to these web sites as there is no particular list on all web sites which states all the houses on sale. If the buyers were to browse the net on their own it would be time consuming as there are lots of information on the internet and it would take ages to get the exact information sort. This would defeat the purpose of on-line shopping, as most buyers prefer on-line information retrieval or shopping, as it is less time consuming and less cost.

Agent technology in e-commerce applications has helped to improve getting on-line information on products to be bought. Agents help increase sales through the better matching of consumer needs with merchant offerings as well as reduce transaction costs through the semi-automation of business (and inter-business) processes.

This project is to set up a system using agent technology, so that it would make it easier for these buyers to get the help of these agents to browse the net for them and get the house which best suit their requirements. All the buyer has to do is fill in a form provided by the agency. In this form the buyer would specify his exact requirement of the house and submit it to the agent on-line. The agent would then search a few search engines simultaneously and get the information that best suit the buyers requirement. The details of the house and the URL of the page that contains this information would be display as the result of the search.

Apart from that the agent would also give its recommendation on which house best suits the buyers requirement. By using agent technology to build the agent, information retrieval for the buyers would be made easier. Thus the practice of on-line shopping would increase in Malaysia.

1.2 Objective

The objective of this project is to develop a system which uses agent technology to provide help to its users to get to the web-sites which has advertisement on the house required by the users. The web site could be on-line sales or just an advertisement with further contact details on how to buy the house. The agent on behalf of the house buyer would search information sort by the clients. The agent would also give its recommendation on which result from the search, which best suits, the buyers needs. The house-agent is built based on the following objectives:

I) To increase the usage of on-line shopping for houses in Malaysia.

II) To improve the standard of on-line shopping in Malaysia.

III) To enable house buyers get the ideal house without having to waste time browsing the net themselves.

IV) To help buyers make right decision as agent would be able to provide the specific information required by the buyers and its recommendation on which house best suits the buyer.

V) To save cost and time for the buyers.

1.3 Scope

The house buyer who seeks the help of the agent would first select the place anywhere in Malaysia where they want the house. These includes information on the state and district.

Next the buyer would let the agent know about the price which would be in a form of certain range such as from RM 60, 000 to RM 70,000 or RM 70,000 to RM 80,000 and so on.

The buyer would have to specify the type of house wanted, that is apartments, condominiums, single story or double story linked house(corner, end or intermediate lot) semi-detached(single or double story) or bungalow(single or double story). After that the buyer would have to specify the size of the house.

Once the filled form is submitted, the agent would talk directly to databases on the Web and deliver the up-to-the-minute details, saving time by searching multiple sites simultaneously. The result would be listed along with the URL to contact regarding the house. The details of the houses would also be displayed for the buyer to judge.

The agent would also give recommendation on which of the result best suit the user. This would be based on the amount of details which best match the buyers need.

CHAPTER 2
A REVIEW ON
AGENT TECHNOLOGY

LITERATURE REVIEW

2.1 AGENT

An agent could be defined as a representative who advises his or her client in a certain area of expertise. Representation and negotiation by a third party has existed ever since people began to communicate with each other. For example long time back in England the official "facilitator" would, for a fee, match up people who wanted audiences with important officials or royalty.

There are many type of agents. An agent could be someone who helps in the purchasing of certain product tangible or non-tangible according to the needs of his or her client. Examples of these agents are insurance agents, real-estate agents, literary agents and many other types of agents.

Agents may represent athletes, writers, models, actors, producers, performers, and other celebrities. They help make their clients' successes happen. However in reality, an agent has very little power to make or break any deal.

Most agents work for a commission, most often a percentage of the sale price of the home being sold. Agents usually don't receive any up-front compensation for their efforts. Instead, they are paid when the sale goes through. If a deal falls apart, the agent makes nothing for his or her efforts.

2.2 AGENT TECHNOLOGY

2.2.1 What are agents?

Agents are computational systems that inhabit some complex dynamic environment, sense and act autonomously in this environment, and by doing so realize a set of goals or tasks for which they are designed[4]

With the sudden boom of agent technology world wide, many researchers have striven to study its impact on society, its implications and applications in domains as diverse as computer games and interactive cinema, information retrieval and filtering, user interface design, electronic commerce and industrial process control. It is also noted that several distributed systems already apply agent technology, and many more commercial and industrial applications are under development by companies world wide[4].

2.2.2 Characteristics of Agents

Autonomous: an agent is able to take initiative and exercise a non-trivial degree of

- Goal-oriented: an agent accepts high-level requests indicating what a
- Collaborative: an agent does not blindly obey commands, but has the
- Flexible: the agent's actions are not "scripted"; it is able to dynamically
- Self-starting: unlike standard programs which are directly invoked by the

Temporal continuity: an agent is a continuously running process, not a "one-shot" computation that maps a single input to a single output, then terminates.

Character: an agent has a well-defined, believable "personality" and emotional state.

Communicative: the agent is able to engage in complex communication with other help in accomplishing its goals.

Adaptive: the agent automatically customizes itself to the preferences of its user based on previous experience. The agent also automatically adapts to changes in its environment.

Mobile: an agent is able to transport itself from one machine to another and across different system architectures and platforms.

2.2.3 Types of agents

The term "agent" is used (and misused) increasingly to describe a broad range of computational entities. Agent document is an entity that resides in environments where it interprets "sensor" data that reflect events in the environment and executes "motor" commands that produce effects in the environment. An agent can be purely software or hardware. In the latter case a considerable amount of software is needed to make the hardware an agent.

This tends to obscure the differences between radically different approaches[9].

- Some agents performs tasks individually ... others need to work together
- Some are mobile ... some static
- Agents communicate via messages ... some don't communicate at all
- Some learn and adapt ... others don't

Agents assist users in a range of different ways:

- they perform tasks on the user's behalf
- they can train or teach the user
- they help different users collaborate
- they monitor events and procedures

Definitions of agent involve some notion of [5]:

- *trust*: the agent will do what you think it will do,
- *personalizability*: the agent can be either learn or be explicitly taught what to do for each individual user, and
- *autonomy*: the agent is allowed to take at least some actions on the user's behalf, without permission or perhaps even notification.

We distinguish between four kinds of agents:

Interface Agents

Computer programs that employ artificial intelligence techniques in order to provide assistance to a user dealing with a particular application ...The metaphor is that of a *personal assistant* who is *collaborating with the user* in the same work environment[6].

Information Agents

"An *information agent* is an agent that has access to at least one, and potentially many information sources, and is able to collate and manipulate information obtained from these sources to answer queries posed by users and other information agents[7].

Commerce Agents

A commerce agent is an agent that provides commercial services (e.g., selling, buying, prices' advice) for a human user or for another agent.

Entertainment Agents

Artistically interesting, highly interactive, simulated worlds ... to give users the experience of living in (not merely watching) dramatically rich worlds that include moderately competent, emotional agents[8].

Despite this diversity, we can identify some common properties. Most uses of agents can be subsumed under two headings[9]:

- Simplifying distributed computing

  Agents as intelligent **resource managers**

- Overcoming user interface problems

  Agents as **personal assistants** which adapt to the user

2.2.4 Importance of agents

The sheer endlessness of the information available through the Internet, which at first glance looks like its major strength, is at the same time one of its major weaknesses. The amounts of information that are at our disposal are too vast: information that is being sought is (probably) available somewhere, but often only parts of it can be retrieved, or sometimes nothing can be found at all. To put it more figuratively: the number of needles that can be found has increased, but so has the size of the haystack they are hidden in. The inquirers for information are being confronted with an information overkill.

2.2.5 Problems with conventional search methods

The current, conventional search methods do not seem to be able to tackle these problems. These methods are based on the principle that it is known which information is available (and which one is not) and where exactly it can be found. To make this possible, large information systems such as databases are supplied with (large) indexes to provide the user with this information. With the aid of such an index one can, at all times, look up whether certain information can or cannot be found in the database, and - if available - where it can be found.

Because of the emergence of information sources such as the world-wide computer network called the internet, everyone in principle can have access to a sheer inexhaustible pool of information. On the Internet this strategy fails completely, the reasons for this being:

The dynamic nature of the Internet itself: there is no central supervision on the growth and development of Internet. Anybody who wants to use it and/or offer information or services on it, is free to do so. This has created a situation where it has become very hard to get a clear picture of the size of the Internet, let alone to make an estimation of the amount of information that is available on or through it

The dynamic nature of the information on Internet: information that cannot be found today, may become available tomorrow. And the reverse happens too: information that was available, may suddenly disappear without further notice, for instance because an Internet service has stopped its activities, or because information has been moved to a different, Unknown location;

The information and information services on the Internet are very heterogeneous: information on the Internet is being offered in many different kinds of formats and in many different ways. This makes it very difficult to search for information automatically, because every information format and every type of information service requires a different approach.

2.2.6 Comparison between search engine features and Agents features

Search Engine feature:	Intelligent Agents can offer:
An information search is done, based on one or more keywords given by a user. This presupposes that the user is capable of formulating the right set of keywords to retrieve the wanted information. Querying with the wrong, too many, or too little keywords will cause many irrelevant information ('*noise*') to be retrieved or will not retrieve (very) relevant information as it	Agents are capable of intelligent search, for instance because tools (such as a thesaurus) enable them to search on related terms as well, or even on concepts. Agents will also use these tools to fine-tune, or even correct user queries (on the basis of a user model, or other user information);

does not contain these exact keywords;	
Information mapping is done by gathering (meta-)information about information and documents that are available on the Internet. This is a very time-consuming method that causes a lot of data traffic, it lacks efficiency (there are a lot of parties that use this method of gathering information, but they usually do not co-operate with others which means that they are reinventing the wheel many times), and it does not account very well for the dynamic nature of the Internet and the information that can be found on it;	Individual user agents can create their own knowledge base about available information sources on the Internet, which is updated and expanded after every search. When information (i.e. documents) have moved to another location, agents will be able to find them, and update their knowledge base accordingly. Furthermore, in the future agents will be able to communicate and co-operate with other agents (such as middle layer agents). This will enable them to perform tasks, such as information searches, quicker and more efficient, reducing network traffic. They will also be able to perform tasks (e.g. searches) directly at the source/service, leading to a further decrease of network traffic;
The search for information is often limited to a few Internet services, such as the WWW. Finding information that is offered through other services (e.g. a 'Telnet-able' database),	Agents can relief their human user of the need to worry about "clerical details", such as the way the various Internet service have to operated. Instead, he or she will only have to

often means the user is left to his or her own devices;	worry about the question what exactly is being sought (instead of worrying about where certain information may be found or how it should be obtained). The user's agent will worry about the rest;
Search engines cannot always be reached: the server that a service resides on may be 'down', or it may be too busy on the Internet to get a connection. Regular users of the service will then have to switch to some other search engine, which probably requires a different way to be operated and may offer different services;	As a user agent resides on a user's computer, it is always available to the user. An agent can perform one or more tasks day and night, sometimes even in parallel. As looking for information on the Internet is such a time-consuming activity, having an agent do this job has many advantages, one of them being that an agent does not mind doing it continuously. A further advantage of agents is that they can detect and avoid peak-hours on the Internet;
Search engines are domain-independent in the way they treat gathered information and in the way they enable users to search in it. Terms in gathered documents are lifted out of their context, and are stored as a mere list of individual keywords. A term like	Software agents will be able to search information based on contexts. They will deduce this context from user information (i.e. a built-up user model) or by using other services, such as a thesaurus service. See chapter four and six for more detailed information about this;

"information broker" is most likely stored as the two separate terms "information" and "broker" in the meta-information of the document that contains them. Someone searching for documents about an "information broker" will therefore also get documents where the words "information" and "broker" are used, but only as separate terms (e.g. as in "an introductory *information* text about stock *brokers*");	
The information on Internet is very dynamic: quite often search engines refer to information that has moved to another, unknown location, or has disappeared. Search engines do not learn from these searches, and they do not adjust themselves to their users. Moreover, a user cannot receive information updates upon one or more topics, i.e. perform certain searches automatically at regular intervals.	User agents can adjust themselves to the preferences and wishes of individual users. Ideally this will lead to agents that will more and more adjust themselves to what a user wants and wishes, and what he or she is (usually) looking for, by learning from performed tasks (i.e. searches) and the way users react to the results of them. Furthermore, agents are able to continuously scan the Internet for (newly available) information about topics a user is interested in.

2.3 E-Commerce

Electronic commerce means doing business online or selling and buying products and services through Web storefronts. The strategic deployment of computer-mediated business tools and information technologies to satisfy business objectives. The term "electronic commerce" has evolved from its meager notion of electronic shopping to mean all aspects of business and market processes enabled by the Internet and the World Wide Web technologies.

Products being traded may be physical products such as used cars or services (e.g. arranging trips, online medical consultation, and remote education). Increasingly, they include digital products such as news, audio and video, database, software and all types of knowledge-based products. It appears then electronic commerce is similar to catalog shopping or home shopping on cable TV.

The term "electronic commerce" has evolved from its meager notion of electronic shopping to mean all aspects of business and market processes enabled by the Internet and the World Wide Web technologies. The best definitions view electronic commerce as a strategy to support the total delivery of products and services to the customer, rather than just another set of tools and technologies. electronic commerce offers fundamentally new ways of doing business, rather than mere extensions of existing practices. It is, in the end, the strategic deployment of computer-mediated business tools and information technologies to satisfy business objectives.

Electronic Commerce As Online Selling. Narrowly defined, electronic commerce means doing business online or selling and buying products and services through Web storefronts. Products being traded may be physical products such as used cars or services (e.g. arranging trips, online medical consultation, and remote education). Increasingly, they include digital products such as news, audio and video, database, software and all types of knowledge-based products. It appears then electronic commerce is similar to catalog shopping or home shopping on cable TV.

Electronic Commerce As a Market. Electronic commerce is not limited to buying and selling products online. For example, a neighborhood store can open a Web store and find the world in its door step. But, along with customers, it will also find its suppliers, accountants, payment services, government agencies and competitors online. This online or digital partners demand changes in the way we do business from production to consumption, and they will affect companies who might think they are not part of electronic commerce. Along with online selling, electronic commerce will lead to significant changes in the way products are customized, distributed and exchanged and the way consumers search and bargain for products and services and consume them. In short, the electronic commerce revolution is in its **effects on processes**. Process-oriented definition of electronic commerce offers a broader view of what electronic commerce is. Within-business processes (e.g. manufacturing, inventorying, corporate financial management, operation), and business-to-business processes (e.g. supply-chain management, bidding) are affected by the same technology and network as are business-to-consumer processes. Even government functions, education, social and political processes undergo changes.

On the surface, the electronic marketplace appears to be something of a perfect market, where there are numerous, worldwide sellers and buyers, who in turn have bountiful information about the market and products, and where no intermediaries are necessary. Such a market is very competitive and efficient (with no need to regulate or intervene arbitrarily).

However, closer looks indicate that consumer searches are not very efficient (due to the cost of having a complete, easily searchable database, and because sellers may not provide all information necessary). Although wholesalers and retail outlets may not be needed, other types of intermediaries appear to be essential for the electronic market to function adequately (e.g. certification authorities, electronic malls who guarantee product quality, mediators for bargaining and conflict resolution, etc.). All these brokers add transaction costs.

Will prices be lower? Digital products are highly customizable due to its **transmutability**, i.e. easy to revise, reorganize and edit. With information about consumer tastes, products will be differentiated (or "customized", e.g. custom news). The number of potential sellers may be low, or even only one, in a highly **differentiated and segmented** market, and the price will tend to approach the maximum price the buyer is willing to pay. (In economic terms, sellers practice "first degree or perfect" price discrimination, which is exact opposite to the result we get in a perfectly competitive market.)

Business-centered electronic commerce began more than two decades ago with the introduction of electronic data interchange (EDI) between firms (sending and receiving order, delivery and payment information, etc.) Even consumer-oriented electronic commerce has a rather long history: each time you use automatic teller machines or present your credit cards, you transact business electronically. These EDI and ATM, however, operate in a closed system; they are of a more convenient communications medium, strictly between the parties allowed in.

2.4 AGENT MEDIATED E-COMMERCE

2.4.1 Properties of agents and agent systems that are useful in commerce applications

Software agents are programs to which one can delegate (aspects of) a task. They differ from "traditional" software in that they are personalized, continuously running and semi-autonomous. These properties make agents useful for a wide variety of information and process management tasks. It should come as no surprise that these same qualities are particularly useful for the information-rich and process-rich environment of electronic commerce[3].

2.4.2 Challenges electronic commerce pose to agent technology

Electronic commerce poses a diverse array of challenges and opportunities for agent systems. From a Consumer Buying Behavior perspective, we see agents playing roles in three primary CBB stages: Product Brokering, Merchant Brokering, and Negotiation corresponding to what to buy, who to buy it from, and how to determine the terms of the transaction respectively. It is in these roles that agent system designers have the challenge to accurately identify and the opportunity to prescribe innovative ways of doing business. Of course, there are other opportunities for agents in e-commerce such as helping automate supply chain management and numerous other back-office tasks.

2.4.3 Requirements which must be met before we can expect to see fielded systems

The requirements depends on the problem and market, of course. We already see agent systems playing visible roles in retail markets (e.g., Firefly, Jango, Dell's computer configurators, etc.) as well as stock markets (e.g., OptiMark Technologies, E-Trade, etc.) However, for agent systems really to become ubiquitous, some pieces need to be in place first.

The two general goals of e-commerce is (1) interoperation and (2) automation. In many cases, there is a dependency of automation upon interoperation. For example, in order to help automate the management of supply chains, there needs to be a semantically interoperable language and protocol for coordinating the parties involved.

Unfortunately, there is currently a lack of common languages and ontology's for e-commerce interoperation.

Although HTML web-scraping may get us by for certain problems (e.g., product information retrieval in retail markets), it is not sufficiently robust to base important business processes upon. As a community with a long history of language and protocol design, we have the opportunity to help define such languages and protocols for e-commerce systems. OPS and ICE are two such examples of this.

Related to this is the need for an open (distributed) registry mechanism which helps tie businesses and e-commerce components together. If not EDI Value-Added Networks (VANs), then we need to find another avenue for managing the evolution of these registries -- e.g., CommerceNet.

There already are agent systems being used to negotiate and make agreements in consumer-to-consumer (e.g., eBay's AuctionWeb) and business-to-business (e.g., FairMarket) and stock markets (e.g., E-Trade and OptiMark Technologies). In order to trust such systems, the agent owners must be assured that the agent will not compromise private information and deviate beyond its constraints. This can occur in the form of insured guarantees (e.g., E-Trade), audits (e.g., TRUSTe), clear controls of strategic and other private information (e.g., OPS/P3P), as well as reputation services (e.g., BizRate and Kasbah's Better Business Bureau).

The implications of agents helping automate negotiations among numerous parties are profound. The first wave will substantially help streamline business-to-business transactions, reducing transaction costs at every stage of the supply chain. At some critical threshold, new types of transactions will emerge in the form of dynamic relationships among previously unknown parties. In this next wave, agents will strategically form and reform coalitions to bid on contracts and leverage economies of scale -- in essence, creating dynamic business partnerships that exist only as long as necessary. The potential is an environment where companies will be at their most agile and marketplaces will approach perfect efficiency.

For certain, consumers must be assured that their agents will not compromise private information and deviate beyond their constraints. This includes account numbers for payment mechanisms. Essentially, consumers must trust their agents and the environment in which agents carry out their tasks.

As agent designers, we can help instill trust in our agents by having them exhibit (mostly) predictable behaviors and having them be able to clearly explain their actions and decisions. In addition, we must be careful not to inflate the users' expectations beyond our agents' capabilities as this can lead to a lack of trust in our systems.

2.4.4 Standards needed to be in place for agent-mediated electronic commerce to be a reality

Security, privacy, communication, and ontology standards need to exist for agents to be most effective. What is also important are processes for interoperating in the face of competing standards (which is the more likely scenario) and distributively managing the evolution of these standards. Standards-by-committee will surely be too slow for the potential dizzying pace of e-commerce.

There is also a need of clear legal jurisdiction on agent-mediated business deals and contracts. For example, what audit trails must agents leave in their digital wakes? What roles will certification authorities, trusted third parties and security/authentication functions play in agent-mediated electronic commerce?

Cryptography and related certification and authentication services will play vital roles in agent-mediated e-commerce. Trusted third-parties will be used as market-makers (e.g., auction "houses") as well as provide related services such as reputation brokering, financial services, escrow services, etc.

Ensuring electronic commerce systems are stable, scalable and have adequate performance times obviously is impossible in an environment as diverse, unpredictable, and unstable as the Internet (as we know it today). (This is one argument in favor of secure EDI VANs.)

Image processing, speech recognition, information filtering and retrieval, and other content-based technologies are slowly making progress to help extract semantics from multimedia. These may one-day help agents find items of interest to their owners or to help accomplish a task.

An alternative approach is to build statistical models of the relationships among multimedia objects (as opposed to their contents). Automated collaborative filtering, for example, enables the recommendation of previously unknown products -- e.g., songs, movies, etc. -- to a consumer through the aggregated ratings of numerous respondents.

2.5 Conclusion

Through out the review of agent mediated e-commerce we have seen how important agent technology is to improve business in a country. At the moment most research and implementation of intelligent agents are done in America. The usage of agent technology in our country is still new and not well versed.

Agent technologies can help realize the elusive one-to-one marketing. Personalization technologies such as automated collaborative filtering and other learning methods can (passively) help consumers identify their needs. This can take the form of well-selected banner ads, or better yet, via more innovative means -- e.g., personalized shopping catalogs, effective cross-sale and up-sale offers, etc.

Agents will also have a large impact economically. Agents will likely help shape the new economy by taking over the roles of some traditional intermediaries (e.g., discount stock brokers) as well as reduce the need of other positions (e.g., customer service and sales). However, agents will also be instrumental in creating new markets (more viable consumer-to-consumer and B-goods). Most importantly, agents will help increase sales through the better matching of consumer needs with merchant offerings as well as reduce transaction costs through the semi-automation of business (and inter-business) processes.

Agents are intelligent and would be more caring towards consumer needs. It is there for necessary for Agent mediated e-commerce in Malaysia. As a developing country it is important to increase the usage of intelligence in electronic business dealing and keep in par with the othe countries.

CHAPTER 3

METHODOLOGY

3.0 INTRODUCTION

A methodology is a proven series of steps and tasks systems developers can follow to build quality systems faster, at lower costs, and with less risk. Many widely accepted methodologies exist, but the basic concepts and practices of a methodology should provide a solid system development foundation. The methodology used to develop the agent should be appropriate to the size, type, and scope of the project.

The project need a life cycle definition and management methodology to achieve consistent, on time, on budget, high quality results in project management and software development

3.1 System Development Life Cycle(SDLC)

A Systems Development Life-Cycle methodology is a structured approach for systems development from systems planning and design through implementation and support [1]. Methodology refers to the set of management processes and procedures applied across the SDLC. Following are the steps followed in SDLC

Step 1: Analysis of the Current System

To develop the new system, an analysis of the current systems using agent technology must be made to determine problems, opportunities and objectives. The goal of systems analysis is to obtain a clear understanding of the system and its shortcomings and to determine opportunities for improvement.

Possible improvement goals of a new system:

Speeding up and streamlining processes

Combining processes

Eliminating redundant input

Reducing redundant output

Cutting costs

Improving security

Some of the methods used to gather information are:

Conduct interviews

Review procedures and policies

Collect sample documents (forms, memos, reports, etc.) Observe operations

A Data-Flow diagram is used to help gather information about the system. It will be designed using a Top-Down method.

Step 2: Define New Requirements

Before development can proceed further, the system must be defined: (example: What is the function of the agent?)

The system requirements are determined: input, storage and output. These can be hardware and/or software. At this point, the availability of hardware and technology and their costs have to be taken into Created by Priyadharshini. Alternatives will be discussed and evaluated.

A systems flowchart will be used at this stage, and a report will be produced. Based on the report, the project will either continue or end. The report can also be used as a basis for bids from vendors of equipment and software.

Step 3: Design the New System

At step 3, the system is designed. How the system is to be constructed is determined. The new system would be designed.

The computer specialist has two goals to meet:

1. Design of the new system: Using design tools, as well as systems flowcharts, program flowcharts, etc., the computer specialist will design the new system.
2. Controls and Security: Controls and security passwords must be established here.

Another report for acceptance is also created based on the design.

Two of the design tools which can be used are:

HIPO

Hierarchy-Input-Processing-Output packages are used to design and document the development of the design and controls.

CASE

Computer Aided Software Engineering tools provide automated support for design which speeds up the design process by providing: graphics tools, reports on file contents and properties, prototypes, quality analysis and programming code.

Step 4: Develop the New System

After the system has been designed and approved, it must be developed. This is when the hardware and software is actually acquired.

Certain decisions must be made. Should the software be bought, or written? Who supplies the hardware? Here open bids for tenders may be made.

In addition, users must be trained on the new system. Sometimes the vendors of the software give training seminars. Also, any new procedures or polices must be taught.

Lastly, the system must be tested. Prototyping is used. The system is tested with minimal prototype data to catch bugs and errors. This is a very time consuming, painstaking process and may take several months.

Step 5: Implement the New System

After the system has been developed and tested, it must be implemented. According to the type of organizations, there are different ways to accomplish this.

There are four main implementation methods:

Direct Implementation

The change to the new system is made all at once. This method is used in a small business or in a larger one where a model was previously developed and thoroughly tested.

Parallel Implementation

Both the old and the new systems are up and running at the same time. When the operation of the new system is satisfactory, the old system is shut down.

Phased Implementation

Implementation of a part of the system and when it is running satisfactorily, another part follows. This is used for extremely large and complicated systems.

Pilot Implementation

If a company has many widely dispersed locations, the new system is implemented in one location at a time. Some of the trained staff would then be moved to the new location.

Step 6: Postimplementation Evaluation and Maintenance

After the system has been implemented, and has been running for a few months, an evaluation is made to determine if it is meeting its objectives.

Maintenance is an ongoing task, not all of which is part of the Systems Development Life Cycle. Changes in government regulations often necessitate periodic changes in the processing requirements of the system. In addition, the system is adjusted and enhanced.

For this phase to be effective, users must have a basic understanding of the Systems Life Cycle to communicate their requirements to the systems analysts. Repeated or major changes then necessitate going back to step one.

Following these steps would make SDLC a proper method to develop the agent. It is a strutcured method and has distinguished phases to help define the requirements and analise and design the new system. Later the development phase is also well planned and is flexible to changes to be made to the method.

3.2 Planning

In the beginning of the project proposal a project schedule was planned out using the Gantt chart.Please refer to Figure 1

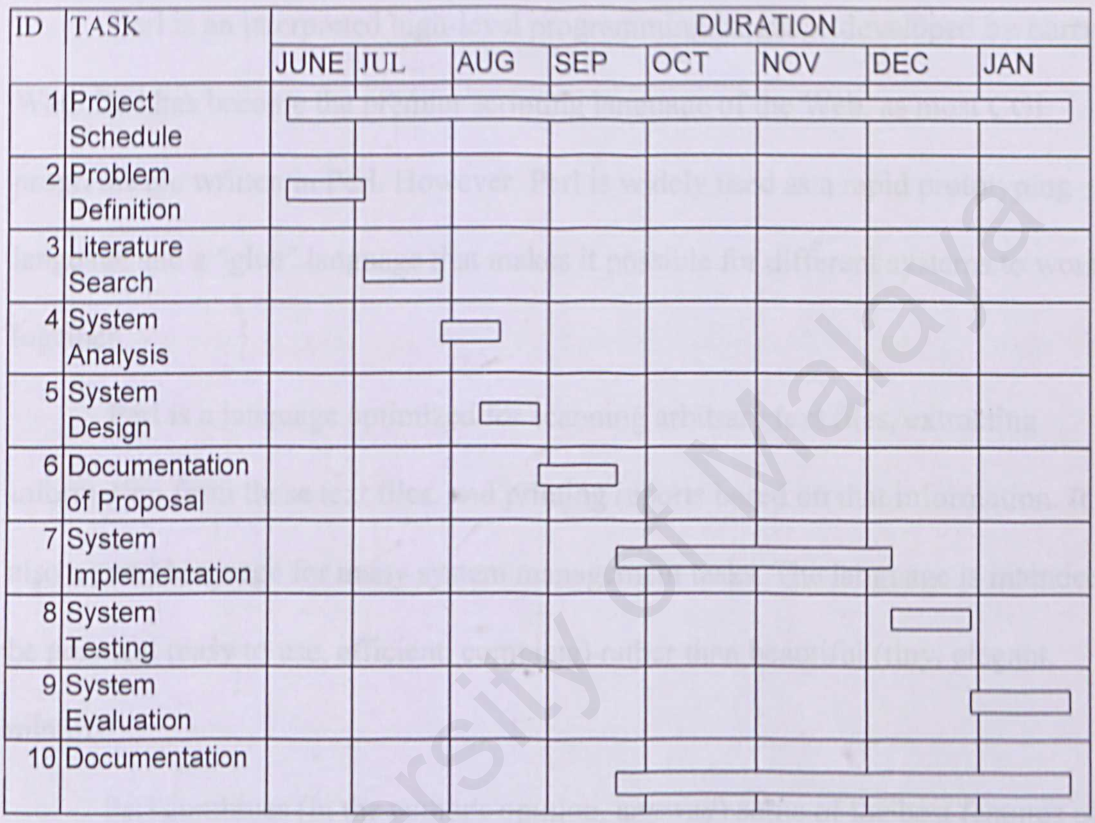


Figure 1 Project Schedule

3.3 TOOLS

3.3.1 Perl

3.3.1.1 Overview and Justification

Perl is an interpreted high-level programming language developed by Larry Wall. Perl has become the premier scripting language of the Web, as most CGI programs are written in Perl. However, Perl is widely used as a rapid prototyping language and a "glue" language that makes it possible for different systems to work well together.

Perl is a language optimized for scanning arbitrary text files, extracting information from those text files, and printing reports based on that information. It's also a good language for many system management tasks. The language is intended to be practical (easy to use, efficient, complete) rather than beautiful (tiny, elegant, minimal).

Perl combines (in the author's opinion, anyway) some of the best features of C, **sed, awk,** and **sh,** so people familiar with those languages should have little difficulty with it. (Language historians will also note some vestiges of **csh**, Pascal, and even BASIC-PLUS) Expression syntax corresponds quite closely to C expression syntax. Unlike most Unix utilities, Perl does not arbitrarily limit the size of the data, if there is memory, Perl can slurp in the whole file as a single string. Recursion is of unlimited depth. And the tables used by hashes (previously called associative arrays) grow as necessary to prevent degraded performance[12].

Perl uses sophisticated pattern matching techniques to scan large amounts of data very quickly. Although optimized for scanning text, Perl can also deal with binary data, and can make dbm files look like hashes. Setuid Perl scripts are safer than C programs through a dataflow tracing mechanism which prevents many stupid security holes.

Perl is popular with system administrators who use it for an infinite number of automation tasks. Perl can be used on a wide range of computing platforms. Because Perl is an interpreted language, Perl programs are highly portable across systems. Many consider Perl as more than a programming language. It is a part of the Internet culture. It is a very creative way of thinking about almost anything[12].

Perl version 5 is nearly a complete rewrite, and provides the following additional benefits:

- Many usability enhancements

  It is now possible to write much more readable Perl code (even within regular expressions). Formerly cryptic variable names can be replaced by mnemonic identifiers. Error messages are more informative, and the optional warnings will catch many of the mistakes a novice might make.

- Simplified grammar

  The new yacc grammar is one half the size of the old one. Many of the arbitrary grammar rules have been regularized. The number of reserved words has been cut by 2/3. Despite this, nearly all old Perl scripts will continue to work unchanged.

- Lexical scoping

  Perl variables may now be declared within a lexical scope, like auto variables in C. Not only is this more efficient, but it contributes to better privacy for programming in the large. Anonymous subroutines exhibit deep binding of lexical variables (closures).

- Arbitrarily nested data structures

  Any scalar value, including any array element, may now contain a reference to any other variable or subroutine. You can easily create anonymous variables and subroutines. Perl manages your reference counts for you.

- Modularity and reusability

  The Perl library is now defined in terms of modules which can be easily shared among various packages. A package may choose to import all or a portion of a module's published interface. Pragmas (that is, compiler directives) are defined and used by the same mechanism.

- Object-oriented programming

  A package can function as a class. Dynamic multiple inheritance and virtual methods are supported in a straightforward manner and with very little new syntax. Filehandles may now be treated as objects.

- Embeddable and Extensible

Perl may now be embedded easily in your C or C++ application, and can either call or be called by your routines through a documented interface. The XS preprocessor is provided to make it easy to glue your C or C++ routines into Perl. Dynamic loading of modules is supported, and Perl itself can be made into a dynamic library.

- POSIX compliant

  A major new module is the POSIX module, which provides access to all available POSIX routines and definitions, via object classes where appropriate.

- Package constructors and destructors

  The new BEGIN and END blocks provide means to capture control as a package is being compiled, and after the program exits. As a degenerate case they work just like awk's BEGIN and END when we use the -p or -n switches.

- Multiple simultaneous DBM implementations

  A Perl program may now access DBM, NDBM, SDBM, GDBM, and Berkeley DB files from the same script simultaneously. In fact, the old dbm open interface has been generalized to allow any variable to be tied to an object class which defines its access methods.

- Subroutine definitions may now be autoloaded

  In fact, the AUTOLOAD mechanism also allows you to define any arbitrary semantics for undefined subroutine calls. It's not for just autoloading.

- Regular expression enhancements

One can now specify nongreedy quantifiers and now do grouping without creating a backreference. It can be used to write regular expressions with embedded whitespace and comments for readability. A consistent extensibility mechanism has been added that is upwardly compatible with all old regular expressions.

- Innumerable Unbundled Modules

  The Comprehensive Perl Archive Network contains hundreds of plug-and-play modules full of reusable code[13].
- Compilability

  While not yet in full production mode, a working perl-to-C compiler does exist. It can generate portable byte code, simple C, or optimized C code.

Perl is (frequently) an interpreted language. This means that you can write your programs and test them without an intermediate compilation step, allowing you to experiment and test/debug quickly and easily. This ease of experimentation flattens the learning curve even more.

Perl almost certainly will provide a viable, and quick solution. Perl is used, quite extensively, and with extremely reliable and valuable results, at many large computer software and/or hardware companies throughout the world. One of the reason is that it is free. In fact, many Unix vendors now ship Perl by default, and support is usually just a news-posting away.

3.5 Common Gateway Interface (CGI)

The Common Gateway Interface (CGI) is a standard for interfacing external applications with information servers, such as HTTP or Web servers. A plain HTML document that the Web daemon **retrieves** is **static**, which means it exists in a constant state: a text file that doesn't change. A CGI program, on the other hand, is **executed** in real-time, so that it can output **dynamic** information. Most CGI applications involve manipulating data in some fashion and accessing external programs and applications. Perl provides easy to use tools that make these tasks easier.

3.6 Perl with WINDOWS 95

Windows 95 is a product of Microsoft which is commonly used as platform by many organizations. It is compatible to Perl version 5. As such windows 95 would be used as platform for this project.

3.4 Proposed System Design

After analyzing the requirements and needs of the new system, the proposal for the agent system design was created. A data low diagram has been done to show the flow of the system. Please refer Figure 2.

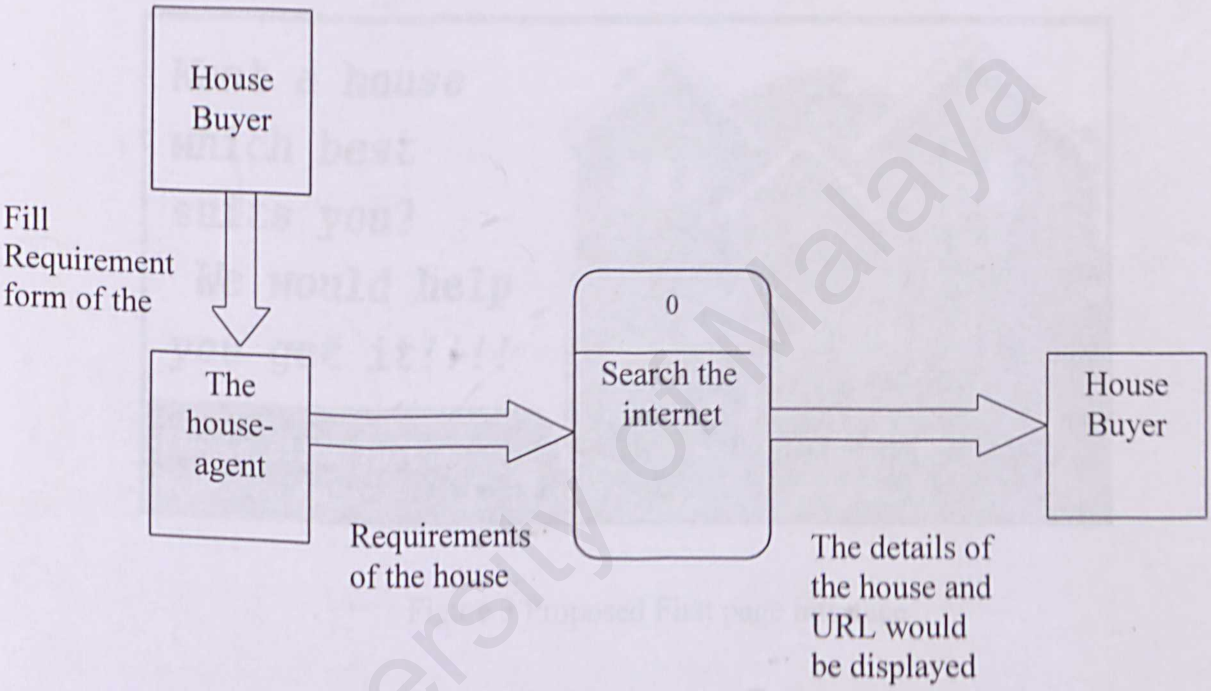

House Buyer
Fill Requirement form of the
The house-agent
Requirements of the house
0
Search the internet
The details of the house and URL would be displayed
House Buyer

Figure 2 Data Flow Diagram of the proposed system

Figure 2 shows how the proposed would work. First the buyer would have to send in their requirements by filling in a form provided by the agent. The details include state, District, Price, Type and Size. Once the form is submitted the Agent would search the entire web for the best match of the requirements. The result of the search would be displayed on-line. The URL of the web site which has the best match to the buyer and the details of the house would be displayed. After that the rest of the deal is between the buyer and the house owner or the real-estate agent. If there were no match The agent would let the buyer know.

3.5 Proposed User Interface Design

Figure 3 shows the proposed user interface of the first page when buyers visit the agent. By clicking on the picture of the house the buyer would be able to enter the next page.

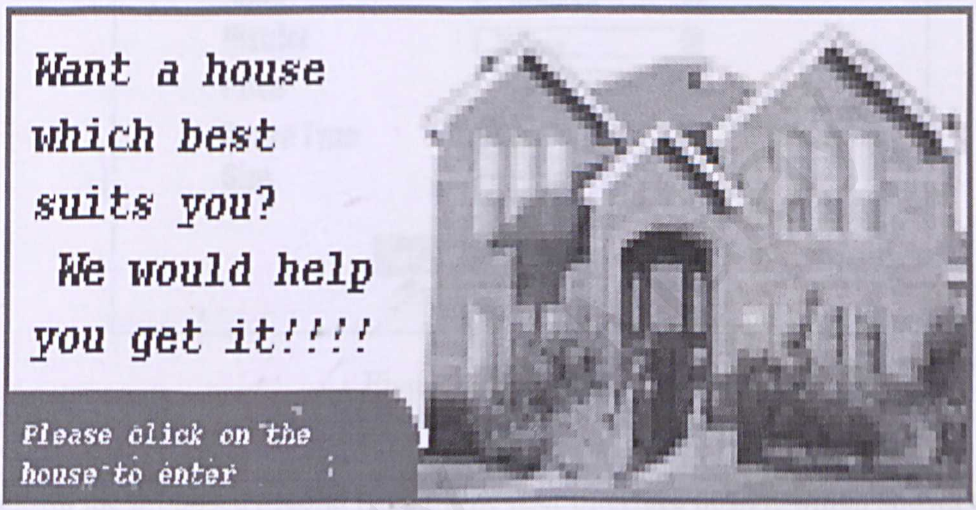
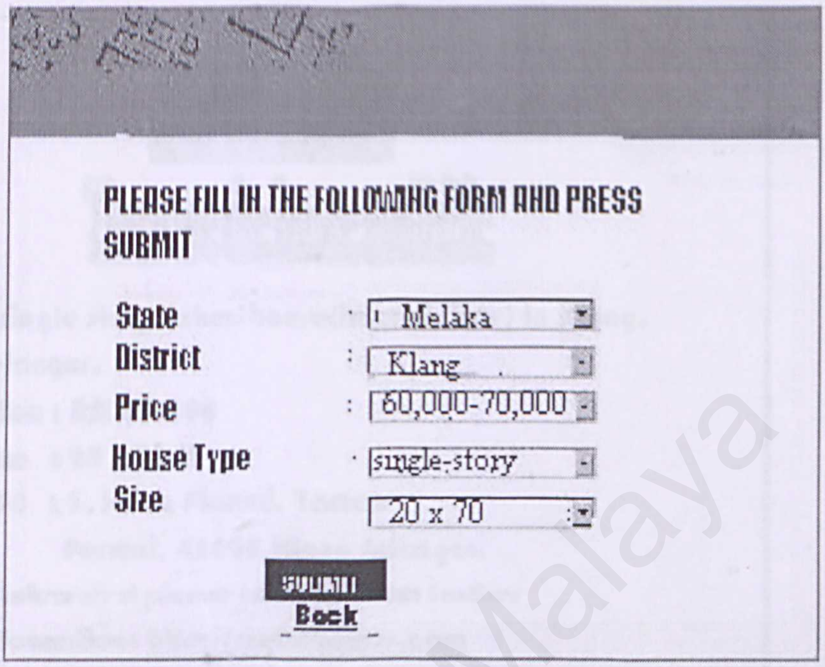

Want a house which best suits you?
We would help you get it!!!!
Please click on the house to enter

Figure 3 Proposed First page interface



PLEASE FILL IN THE FOLLOWING FORM AND PRESS SUBMIT

State : Melaka

District : Klang

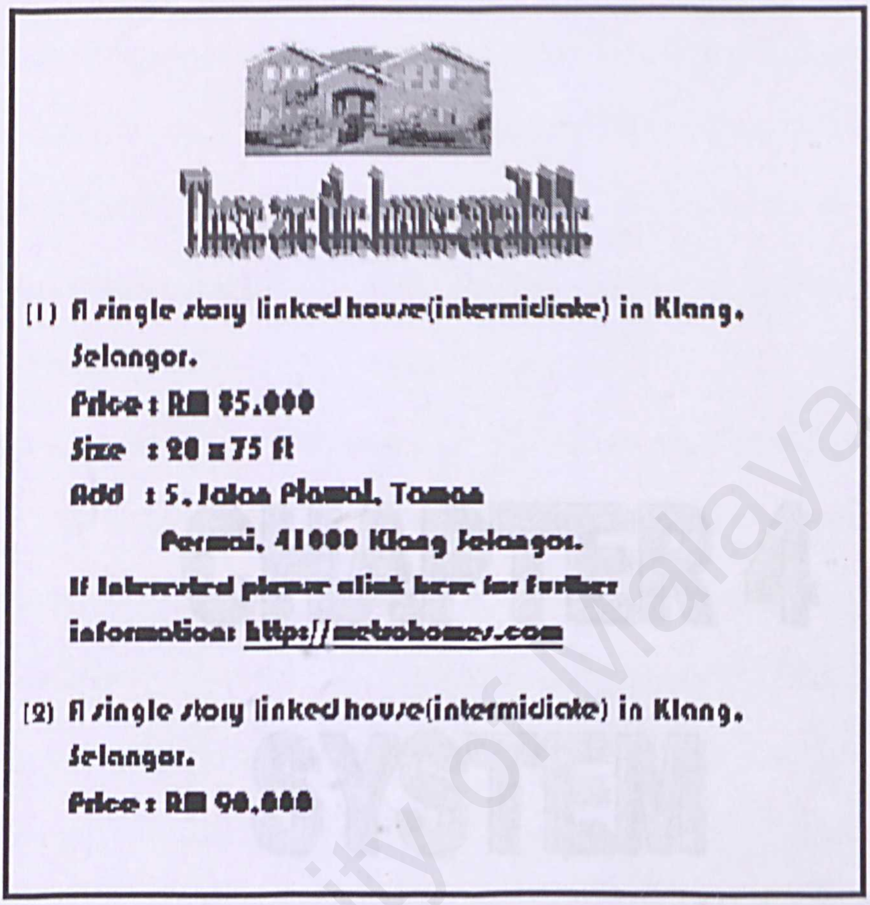
Price : 60,000-70,000

House Type : single-story

Size : 20 x 70

Figure 4 Proposed Input Interface

Figure 4 shows the proposed second page of the user interface design which shows the form to be filled by the buyer. All information are in List down menu so that it would be easier for the buyers to choose what they want. After filling in the form the buyer would have to click on the submit button, and the agent would start searching. The result of the search would be displayed on-line on the screen. Figure 5 shows the proposed interface of the output page.



These are the house available

(1) A single story linked house(intermidiate) in Klang,
Selangor.
Price : RM 85.000
Size : 20 x 75 ft
Add : 5, Jalan Plamal, Taman
Permai, 41000 Klang Selangor.
If interested please click here for further
information: http://metrohomes.com

(2) A single story linked house(intermidiate) in Klang,
Selangor.
Price : RM 90.000

Figure 5 Proposed output interface

CHAPTER 4 SYSTEM IMPLEMENTATION

4.0 INTRODUCTION

System implementation is a process that converts the system requirement and designs into program codes. Purpose of implementation is to program the system, build all data files, test the new system, install system components, covert and cease operation of prior systems, train users and turn over to new owner. The steps involved are coding, testing, conversion, documentation, training and support.

During the development of Mighty House Agent, changes were made regarding the software used to develop Mighty House Agent. The Might House Agent was developed using Visual Interdev 6.0 as the development tool and Active Server Pages (ASP), VBScripts and JavaScript as the programming languages instead of PERL and CGI. During implementation, PERL could not be compiled in personal computer with a Personal Web Server(PWS). As there were not enough facilities to develop the system in such environment, changes had to be made. Section 4.3.2 till 4.3.8 gives a more detail explanation on the software's used.

4.1 DEVELOPMENT ENVIRONMENT

Mighty House Agent was developed using a personal computer with a local web server. Section 4.2 and 4.3 describes the software and the hardware used.

4.2 HARDWARE

Processor	Pentium MMX 166MHs CPU
RAM	32M
Floppy Drive	1.44MB
Hard Disk	1.7 GB
Keyboard	WIN 95 105 keys
Monitor	17” Colour
CD ROM Drive	18 X

4.3 SOFTWARE

4.3.1 Tools

Operating System	Microsoft Windows 98
Programming tool	Microsoft Visual Interdev 6.0
Web Server	Personal Web Server
Interface Designing	Adobe PhotoShop 5.5, Microsoft Paint, Macromedia Dreamweaver 4.0
Web Page Viewing	Microsoft Internet Explorer 3.0

4.3.2 Visual Interdev 6.0

Visual InterDev 6.0 is a project management software for high-end Web development[14]. It integrates many of the existing tools for Web development, and throws in a few hefty tricks of its own for good measure. Its main features are, its support for Microsoft's Active Server Pages, a method for server-side scripts to generate HTML pages, its support for database integration from desktop (Access and MS FoxPro) to high-end (ODBC compatibility) and its support for VBScript and JScript (Microsoft's JavaScript implementation) in HTML files.

Visual InterDev 6.0 has visual design tools, templates and wizards which helps do everything from generating SQL commands with a point-and-click interface to manipulating exposed ActiveX objects. It also has a special version of Microsoft FrontPage for WYSIWYG editing, a color-coded HTML text editor and web project file management and link management tools. Visual InterDev 6.0 also supports VBScript to automate repetitive tasks.

Microsoft Visual Interdev 6.0 provides key features to increase developer productivity—including a shared integrated development environment that helps developers build their solutions faster, regardless of the language used. Other enhancements include easier window management; design tools for Web and Windows forms, components, and XML; and visual database tools.

One of the great features of a Visual InterDev project is the automatic synchroni -zation of local files and the master copies of those files on the Web server. Visual

InterDev automatically creates a copy of files created onto the Web server. This eliminates a particularly error-prone step from your Web development process.

Microsoft Visual Interdev 6.0 includes a variety of visual design tools, wizards, templates, and complete online documentation to help you do everything from creating and generating SQL queries and commands to working with ActiveX controls. Visual InterDev also provides better team- and source-management capabilities.

This well-designed development environment is great for creating and maintaining dynamic Web sites, especially if Microsoft-centric environment is used. Skilled Web builders will find Visual InterDev a great solution and a good foundation upon which to build.

4.3.3 Hyper Text Markup Language (HTML)

HTML stands for Hypertext Markup Language. "Hypertext" refers to the way links on web pages cross-reference other pages, while a markup language consists of a set of text commands which, when interpreted by software (Netscape Navigator or Internet Explorer, for example), result in a visual document[15]. HTML is the markup language that web pages are made with.

4.3.4 Active Server Pages (ASP)

Active Server Pages is a programming environment that gives the ability to generate dynamic HTML pages with the help of server side scripting. VBScript is the default scripting language for ASP, but JavaScript, Perl or any other scripting language can be used for server side scripting in an ASP page. An ASP page is almost the same as a HTM or HTML page, the only difference is that an ASP page has the '.asp' extension. Active Server Page can include both client side and server side scripts. In an ASP page VBScript is usually used as the server side and Java Script as the client side scripting language[16].

This means your ASP file is simply a file that can contain any combination of HTML, scripting, and calls to components. When changes are made on the ASP file on the server, only changes to the file needs to be saved and the next time the Web page is loaded, the script will automatically be compiled. How does this happen? It works because ASP technology is built directly into Microsoft Web servers, and is thus supported on all Microsoft Web servers: Windows NT Internet Information Server (IIS) 3.0, Windows NT Workstation, and Windows 95 Personal Web Server.

ASPs are server-generated pages which can call other programs to do things like access databases, serve different pages to different browsers - basically, anything we used to do with CGI. ASP is almost as efficient as writing code directly to the server's application program interface, and it's a lot more efficient than CGI because it runs as a service and can take advantage of multithreaded architectures.

Active Server Pages is an open, compile-free application environment in which HTML, scripts, and reusable ActiveX server components could be combined to create dynamic and powerful Web-based business solutions.

ASP has evolved into an "open technology framework," meaning there is no need to use Microsoft products to create code in it, though that's still the best way to go, honestly. Nowadays, ASP pages can be created using whatever language, but VBScript is still the most common choice. ASPs, MS developer tools are actually pretty good and are written to save time.

4.3.5 VBScript

Microsoft Visual Basic Scripting Edition (VBScript) is a subset of the Microsoft Visual Basic language. It is implemented as a fast, portable, lightweight interpreter for use in World Wide Web browsers and other applications that use Microsoft ActiveX Controls, Automation servers, and Java applets[17].

4.3.6 JavaScript

JavaScript is a platform-independent, event-driven, interpreted programming language[18] developed by Netscape Communications Corp. and Sun Microsystems. Originally called LiveScript (and still called LiveWire by Netscape in its compiled, server-side incarnation), JavaScript is affiliated with Sun's object-oriented programming language Java primarily as a marketing convenience. They interoperate well but are technically, functionally and behaviorally very different.

JavaScript is useful for adding interactivity to the World Wide Web because scripts can be embedded in HTML files (i.e., web pages) simply by enclosing code in a <SCRIPT> </SCRIPT> tag pair. All modern browsers can interpret JavaScript -- albeit with some irritating caveats. (More about them below.)

In practice, JavaScript is a fairly universal extension to HTML that can enhance the user experience through event handling and client-side execution, while extending a web developer's control over the client's browser.

Java Script is an easy-to-learn programming language which can be built into Web pages, so that it executes from within the browser rather than on the web server.

4.3.7 Adobe Photoshop 5.5

Photoshop's abundance of exquisite bitmap manipulation tools (from brushes to color correction) make it the reigning monarch of image editing. Photoshop has matured into a program that can handle both paper and Web output with amazing grace. Photoshop also has easier to use text controls; improved workflow management and many Web graphics tools.[19]

4.3.8 Macromedia Dreamweaver 4.0

Dreamweaver, Macromedia's professional visual Web editor, has often been lauded for its easy-to-use but powerful visual editing environment. Dreamweaver 4.0 shifts the focus a bit with improvements to its coding environment. The features are simple, yet they will be helpful to developers who like to hand-code much of the time but who use Dreamweaver for its ability to design complex tables and place layers or to rearrange a page's layout visually

Dreamweaver 4.0 still provides the HTML Inspector for hand-coding, but it is better integrated within the program because it could launch within the main window.

One of the best parts of Dreamweaver is its integration with other Macromedia products. Working with other technologies is easy enough, as well. Any non-HTML file opens in the Code view automatically so that its source code is not altered. Dreamweaver 4.0 also has a new JavaScript Debugger to help you execute and fix any script errors[20].

All in all, Dreamweaver continues to be one of the very best visual editing tools on the market. Although its newest features don't seem groundbreaking in themselves, it's the overall emphasis on hand-coding that makes Dreamweaver an even more well-rounded program.

4.4 PROGRAMMING

4.4.1 Methodology

The methodology used is modular approach. Modular approach is useful, as it would make it easier for future modification and enhancement.

4.4.2 Modular Programming

As programs get larger, they get hard to manage Design, Coding and testing. Breaking a program into smaller pieces allows us to deal with each piece individually. The advantages of modular programming are as the following:

1) Modules allow us to design each task separately, without worrying about what other tasks have to do

2) Modules allow us to test each task as it is coded, without waiting until the whole program is complete

3) Modules make it easier for several people to work on different parts of the same program.

4.4.3 Types of Modules

The modules used in Mighty House Agent are as following:

1) Interface

In the interface module all the interface was developed. All pages has Mighty House Agent banner on the top and links to other pages on the side

2) Search

Performs the search process according to the keyword typed in by the user.

4.4.4 Coding Principles

The coding principles used in the development of Mighty House Agent are Coding conventions such as program labeling, naming convention, comments and indentation. Readability was given importance to enable future enhancement. A part from that the other principle used is Maintainability. This made it easier to be corrected and revised.

4.4.5 Web Page Development

Mighty House Agent is a web application, which make use of internet browser. As such it is coded using HTML, VBScripts and JavaScripts. For both client and server side scripting Active Server Pages(ASP) is used. ASP files are processed by the web server before forwarding to the browser. The user will not be able to use the

ASP codes. The ASP codes are interpreted into HTML codes by the web server whenever user requests the ASP files.

ASP is an ideal development tool for Mighty House Agent because it completely integrates with HTML files. Apart from that ASP applications are easy to create without the need for compiling or linking of programs. ASP allows the development of interactive business application.

CHAPTER 5

SYSTEM TESTING

CHAPTER 5
SYSTEM TESTING

5.0 INRODUCTION

Testing is a process of verification and validation of a developed system. The objective of testing is to find error. According to many researchers a good test is when a test is able to produce many error in the system.

5.1 System testing Objectives

Establishing Software testing objectives was a critical part of planning the system testing process. Defining testing objectives was also one of the most difficult tests planning activities because humans frequently do not have a clear idea of what they want to do until they begin to do it. As such the test plans changed during test process execution.

The testing objective for Mighty House Agent was to enssure

1) there were no future execution problems.

2) all icons are able to connect to the relevant page.

3) as much as possible to reduce the possibility of wrong input by the customer

4) a wrong input by the user is notified.

5) there were no grammatical error

5.2 Types Of System Tests carried out on Mighty House Agent

Three types of System testing was carried on Mighty House Agent, System Verification Testing, Customer Verification Testing, and Customer Validation Testing. The first two kinds of System Tests are designed to verify that the system does meet its design objectives. They are destructive in nature and intended to find areas where the system does not accomplish its objectives so that corrections can be made.

System verification test carried out was to test if Mighty House Agent met its objective of searching according to the keyword entered by the customers. The test was conducted by entering different values. Each time an error was found, the error is corrected and the test is carried out again, until the error could be rectified.

Customer Validation Testing is to demonstrate that the system works. Customer Verification Test was of the same manner system verification testing but it was conducted by a few selected people representing the future customers. These selected people were asked to test the system and errors detected were corrected and the test process was repeated.

The third type that is system validation testing is designed to validate the system and is intended to be a positive (confidence building) experience demonstrating how the system fulfills its requirements. *Validation* means that something actually works as it was intended to work. If this is so, then the product being tested meets its requirements.

System validation testing was conducted based on the project objective set in the beginning of the development of Mighty House Agent. Each objective was tested to check if the requirements were met.

CHAPTER 6
SYSTEM EVALUATION

6.0 INTRODUCTION

The development of Mighty House Agent was not a very easy task. The testing process revealed some problems which made it difficult to proceed smoothly. Some improvisation was done but only within the pre-specified limits. As such some of the problems were solved and some were not solved .

The following section explains on the problems encountered and its solutions, the strengths and limitation of Mighty House Agent and as well as suggestions for future enhancement.

6.1 SYSTEM STRENGTH

6.1.1 User Friendly

Mighty House Agent is user friendly as it has listbox, which provides options for users to select from, so that the user has an idea of what they are looking for. Mighty House Agent is also colourful and has attractive Graphical User Interface and this would encourage more users to explore Mighty House Agent and use their services. Mighty House Agent also has instructions as in how to use their web page and their services to makes sure the users do not get confused.

6.1.2 Searching Capabilities

Mighty House Agent is capable of searching for the house wanted by the customer based on five categories. The customer could choose to search for the place they want their house either from state name or district/area name. They could search for the house based on the size, type and price according to their needs.

6.1.3 Saves Time and Effort

Mighty House Agent is a web site, which helps Malaysians get the appropriate house for themselves in an easier way. It saves a lot of time and effort compared to the traditional way of approaching an real-estate agent and so forth. This way one just has to click the button and he gets to the exact house, which fulfills his needs. One does not have to go hunting for houses from one place to the other.

6.1.4 Savings from Marketing Tools

Mighty House Agent also is more cost saving compared to the traditional way as those interested in selling their house has to just submit their particulars to Mighty House Agent and they would display it for them. As for real estate agents, Mighty House Agent is cost saving because there are less need to spend on marketing tools such as advertisement and so forth.

6.1.5 Links to Other Sites

Mighty House Agent also provides their viewers with links with other real-estate agency web sites, Malaysian search engines and other search engines. These links would be of great help for those interested in buying a house.

6.2 PROBLEMS ENCOUNTERED AND SOLUTIONS

6.2.1 Change of Software

The software tools used to develop Mighty House Agent had to be changed from the proposed software that is PERL to Visual Interdev 6.0 because there was not enough facilities to develop Mighty House Agent in a environment where a separate server could upload Perl and compile it. Thus Mighty House Agent had to be developed in a personal computer with Personal Web Server and the appropriate software had to chosen to replace PERL. ASP was chosen as it had better advantage in this environment.

6.2.2 Insufficient Knowledge

The development of Mighty House Agent was not an easy task as the technology used was Agent Technology, which is still very new in Malaysia and guidance to build an agent was limited. The Mighty House Agent actually is capable of searching but it does use the proper agent technology. Thus it was not able to meet some of the proposed requirement of the system.

6.2.3 Determining Project Scope

Determining the project scope was another problem faced during the development of Mighty House Agent. The scope has been limited to just houses instead of all sort of properties such as land lots, factories, shop lots and so forth.

6.3 SYSTEM LIMITATION

6.3.1 No Full Utilization of Agent Technology

Mighty House Agent is not an application fully based on agent technology in e-commerce. Searching for a house is not based on meeting all the requirements such as state, district, price, type and size for a house together. It is treated separately and the search is done one by one.

6.3.2 Customers Wanting to Advertise Their House Need Good Knowledge of E-mailing .

In this sense, Mighty House Agent is not user friendly because those with no knowledge on how to send attachments would find it difficult to send in the information required to advertise their houses.

6.3.3 There is no administrator module

Each time a house is loaded into the web site, it must be coded using HTML. Thus the administrator has to be well versed with HTML

6.3.4 No Online E-Commerce Facilities

Mighty House Agent does not provide its customers with on-line e-commerce facilities such as direct buying and selling of houses and cash transactions which would involve banks as well. Apart from that the services provided are not charged on-line either.

6.3.5 No Online Help

Mighty House Agent has no online help where if users of the web site are stuck half way while trying to get to their ideal house. They could only e-mail and wait for the reply from the web master/administrator.

6.4 ENHANCEMENTS

6.4.1 Add In More Agent Technology Features

Mighty House Agent should be able to search for a house based on all the requirements and give the customer the best house which suit the customers needs.

6.4.2 Provide A Form For those Interested in Selling/Advertising

Those interested in advertising in Mighty House Agent would just have to fill in a form an submit it to the administrator instead of e-mailing it.

6.4.3 Provide Online help

Mighty House Agent could be improved by providing online help, where if one has to know something he could be given the details needed online by the administrators of Mighty House Agent

6.4.4 Provide Extra Services

Mighty House Agent could provide other services such as e-commerce facilities and loan calculation. This would attract more customers to use Mighty House Agent.

CHAPTER 7
CONCLUSION

7.0 CONCLUSION

The Mighty House Agent on whole has met the objective of providing housing service to its customers. It is a system where one can find the perfect house for him/her without having to put in a lot of effort. It saves a lot of time as well.

Mighty House Agent uses agent technology to search for the house according to the place, size, type or price. However agent technology is not fully utilized in Mighty House Agent.

Mighty House Agent has its own strengths such as attractive user interface and searching capabilities. There are certain limitations as well. Not all objectives of developing Mighty House Agent were met in the final.

Mighty House Agent on the whole is a starting point to improve on e-commerce usage in Malaysia and help Malaysia become more advanced in the field of Information Technology. Suggested enhancement towards Mighty House Agent would make Mighty House Agent a perfect housing agent.

REFERENCE

REFERENCE

[1] Computer Concepts

http://cs104.cs.uwindsor.ca/develop.htm

Date referred : 11September 2000

[2] CGI : Common Gateway Interface

http://hoohoo.ncsa.uiuc.edu/cgi/intro.html

Date referred : 2 July 2000

[3] H. Nwana, J. Rosenschein, T. Sandholm, C. Sierra, P. Maes, and R. Guttman.

"Proceedings [illegible] Trading

(AMET'98)"

Minneapolis, Minnesota, May [illegible]

http://ea.www.media.mit.edu/[illegible].html

Date referred : [illegible] 2000

[4] Pattie Maes, MIT Media Lab Software Agents Group

http://www.geocities.com/[illegible]/

Agents_definition.html

Date referred : 8 August 2000

REFERENCE

[1] Computer Concepts

http://cs104.cs.uwindsor.ca/develop.htm

Date referred : 11September 2000

[2] CGI : Common Gateway Interface

http://hoohoo.ncsa.uiuc.edu/cgi/intro.html

Date referred : 2 July 2000

[3] H. Nwana, J. Rosenschein, T. Sandholm, C. Sierra, P. Maes, and R. Guttman,

"*Proceedings of the Workshop on Agent Mediated Electronic Trading (AMET'98)"*

Minneapolis, Minnesota, May 1998.

http://lcs.www.media.mit.edu/people/guttman/research/pubs/agents98.html

Date referred : 8 August 2000

[4] Pattie Maes, MIT Media Lab,Software Agents Group

http://www.geocities.com/ResearchTriangle/Thinktank/4633/

Agents_definition.html

Date referred : 8 August 2000

[5] Lenny Foner

http://foner.www.media.mit.edu/people/foner/agents.html

Date referred : 5 September 2000

[6] O. Etzioni(ed.)

Software Agents *" Papers from the 1994 Spring Symposium (Technical Report SS-94-03), pp.71-78*," AAAI Press.

Date referred : 30 August 2000

[7] M. Wooldridge and N. Jennings

"*Intelligent agents: theory and practice*", in The Knowledge Engineering Review, Vol.10:2, 1995, 115-152

Date referred : 23 July 2000

[8] Bates. J., Bryan Loyall A. and Scott Reilly W

"Integrating reactivity, goals and, emotion in a broad agent", 1992

Date referred : 18 August 2000

[9] Michael Weiss - MITEL Corp

"A gentle introduction to agents and their applications".

http://www.magma.ca/~mrw/agents/what.html

[10] Pattie Maes

"Agents that Reduce Work and Information Overload"

http://pattie.www.media.mit.edu/people/pattie/CACM-94/CACM-94.p1.html

Date referred : 19 August 2000

[11] Pattie Maes with Robert Guttman, Alexandros Moukas, Giorgos Zacharia

"*Software Agents and the Future of Electronic Commerce*" Software Agents Group, MIT Media Laboratory.

http://pattie.www.media.mit.edu/people/pattie/ECOM/sld033.htm

Date referred : 29 July 2000

[12] Http://www.perl.com/

[13] Perlmodlib manpage

http://www.perl.com/CPAN

Date referred : 20 August 2000

[14] http://www.webdeveloper.com/html/html_reviews_visdev.html

Date referred : 23 November 2000

[15] Introduction to HTML

http://www.w3.org/MarkUp/Activity

Date referred : 23 November 2000

[16] Introduction to Active Server Pages

http://hotwired.lycos.com/webmonkey/98/39/index2a.html

[17] http://msdn.microsoft.com/scripting/default.htm?/scripting/vbscript

Date referred : 15 November 2000

[18] http://idm.internet.com/faqs/jsfaq/gen1.html

Date referred : 15 November 2000

[19] Review on Adobe Photoshop

http://www.cnet.com/software

Date referred : 2 January 2000

[20] Review on Macromedia Dreamweaver 4.0

http://www.webdeveloper.com/html/html_reviews_visdev.html

Date referred : 29 July 2000

APPENDIX

USER MANUAL

TABLE OF CONTENTS

CHAPTER 1: GETTING STARTED WITH MIGHTY HOUSE AGENT

1.1 SYSTEM INSTALLATION

The installation of Mighty House Agent is not needed as it is already in the personal web server.

1.2 HARDWARE REQUIREMENT

These are the minimum hardware requirement for the Mighty House Agent to work

- Pentium 166MHs CPU
- 16M RAM
- 1.44MB Floppy Drive
- 1.7 GB Hard Disk
- 18 X CD ROM Drive
- WIN 95 105 keys Keyboard
- 17" Colour Monitor

1.3 SOFTWARE REQUIREMENT

▪ Operating System	-	Microsoft Windows 98
▪ Web Server	-	Personal Web Server
▪ Web Browser	-	Microsoft Internet Explorer 3.0 and above

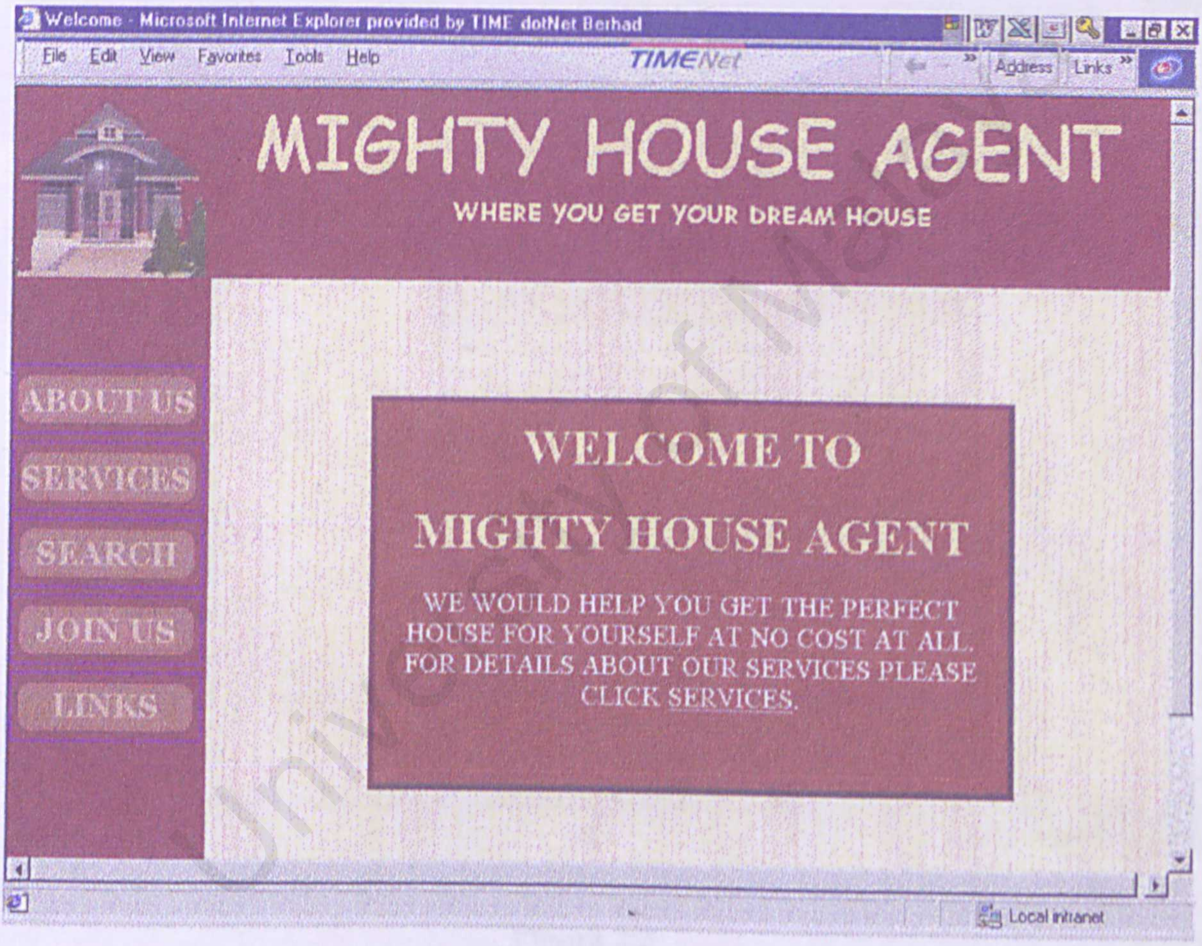
1.4 HOW TO ACCESS MIGHTY HOUSE AGENT WEB SITE

1) Open the internet explorer
2) Type in the URL http://localhost/Project4/ASP/welcomepage.asp
3) The first page of Mighty House Agent would be displayed.

CHAPTER 2: FUNCTIONS OF MIGHTY HOUSE AGENT

2.1 WELCOME PAGE OF MIGHTY HOUSE AGENT

The welcome page of Mighty House Agent gives a brief description of what Mighty House Agent is all about. Figure 2.1 shows the welcome page of Mighty House Agent.


Welcome - Microsoft Internet Explorer provided by TIME dotNet Berhad
File Edit View Favorites Tools Help
TIMENet
Address
Links
MIGHTY HOUSE AGENT
WHERE YOU GET YOUR DREAM HOUSE
ABOUT US
SERVICES
SEARCH
JOIN US
LINKS
Local intranet

WELCOME TO

MIGHTY HOUSE AGENT

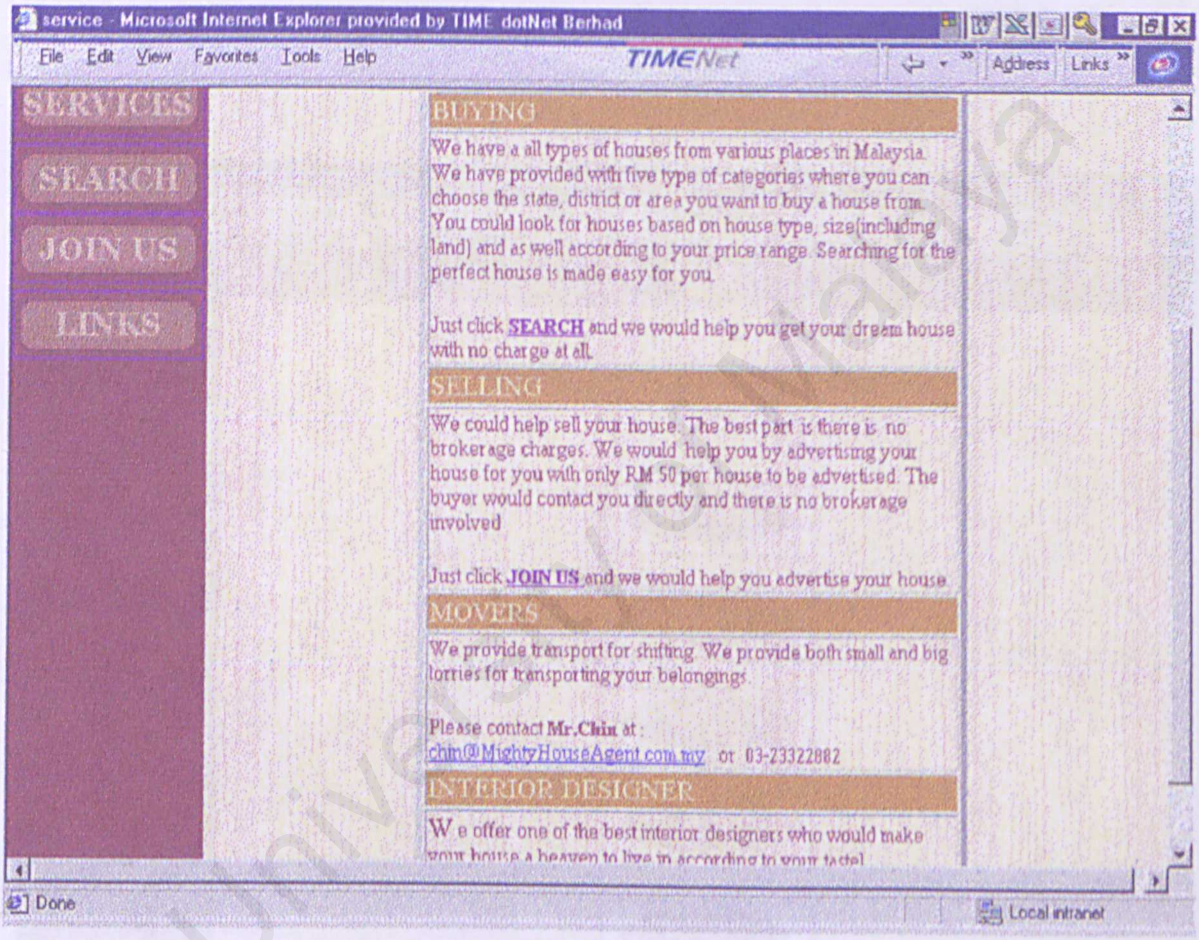
WE WOULD HELP YOU GET THE PERFECT HOUSE FOR YOURSELF AT NO COST AT ALL. FOR DETAILS ABOUT OUR SERVICES PLEASE CLICK SERVICES.

Figure 2.1

The side division of the interface shows five buttons that would redirect the page to the another page once clicked. All pages in Mighty House Agent have these five buttons at the side. The **ABOUT US** button would display the same page as the welcome page.

2.2 TO VIEW SERVICES PROVIDED BY MIGHTY HOUSE AGENT

The **SERVICES** button would display all the services provided by Mighty House Agent and the ways to get these services. Figure 2.2 shows this page.


service - Microsoft Internet Explorer provided by TIME dotNet Berhad
File Edit View Favorites Tools Help
TIMENet
Address
Links
SERVICES
SEARCH
JOIN US
LINKS
Done
Local intranet

BUYING

We have a all types of houses from various places in Malaysia. We have provided with five type of categories where you can choose the state, district or area you want to buy a house from. You could look for houses based on house type, size(including land) and as well according to your price range. Searching for the perfect house is made easy for you.

Just click **SEARCH** and we would help you get your dream house with no charge at all.

SELLING

We could help sell your house. The best part is there is no brokerage charges. We would help you by advertising your house for you with only RM 50 per house to be advertised. The buyer would contact you directly and there is no brokerage involved

Just click **JOIN US** and we would help you advertise your house

MOVERS

We provide transport for shifting. We provide both small and big lorries for transporting your belongings.

Please contact **Mr.Chin** at :
chin@MightyHouseAgent.com.my or 03-23322882

INTERIOR DESIGNER

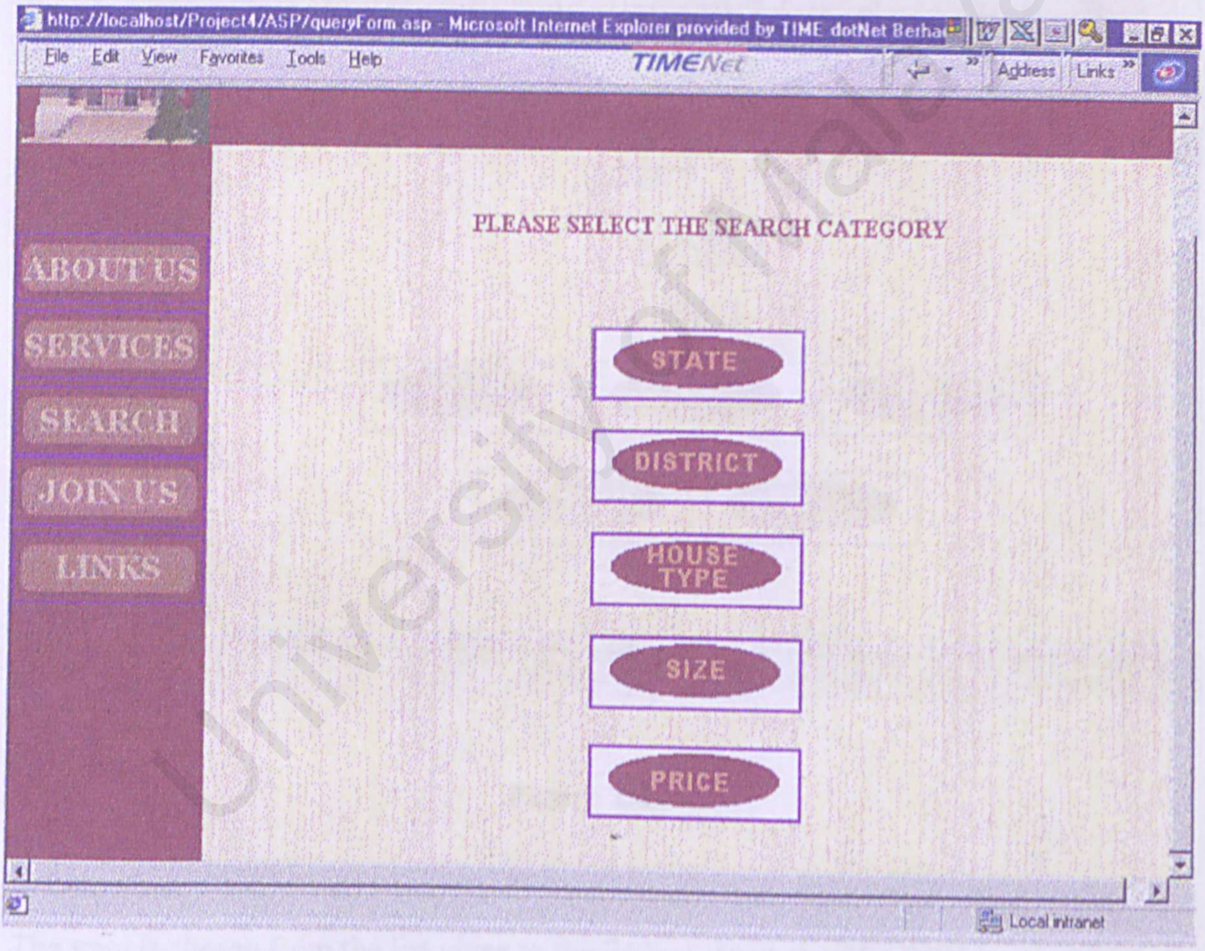
W e offer one of the best interior designers who would make your house a heaven to live in according to your taste!

Figure 2.2

The services provided by Mighty House Agent are Buying(search), Selling, Movers and Interior Designers.

2.3TO SEARCH FOR HOUSES

The **SEARCH** button would display a page similar to Figure 3. The page would show the five categories where one could search for his/her house. The categories are displayed in the form of button with STATE, DISTIRICT, HOUSE TYPE, SIZE and PRICE written on it. Figure 2.3 clearly shows this.



http://localhost/Project4/ASP/queryForm.asp - Microsoft Internet Explorer provided by TIME dotNet Berhad

PLEASE SELECT THE SEARCH CATEGORY

Figure 2.3

2.3.1 SEARCH BY STATE

When the **STATE** button from the search page(Figure 2.3) is clicked a page similar to Figure 2.4 is displayed.

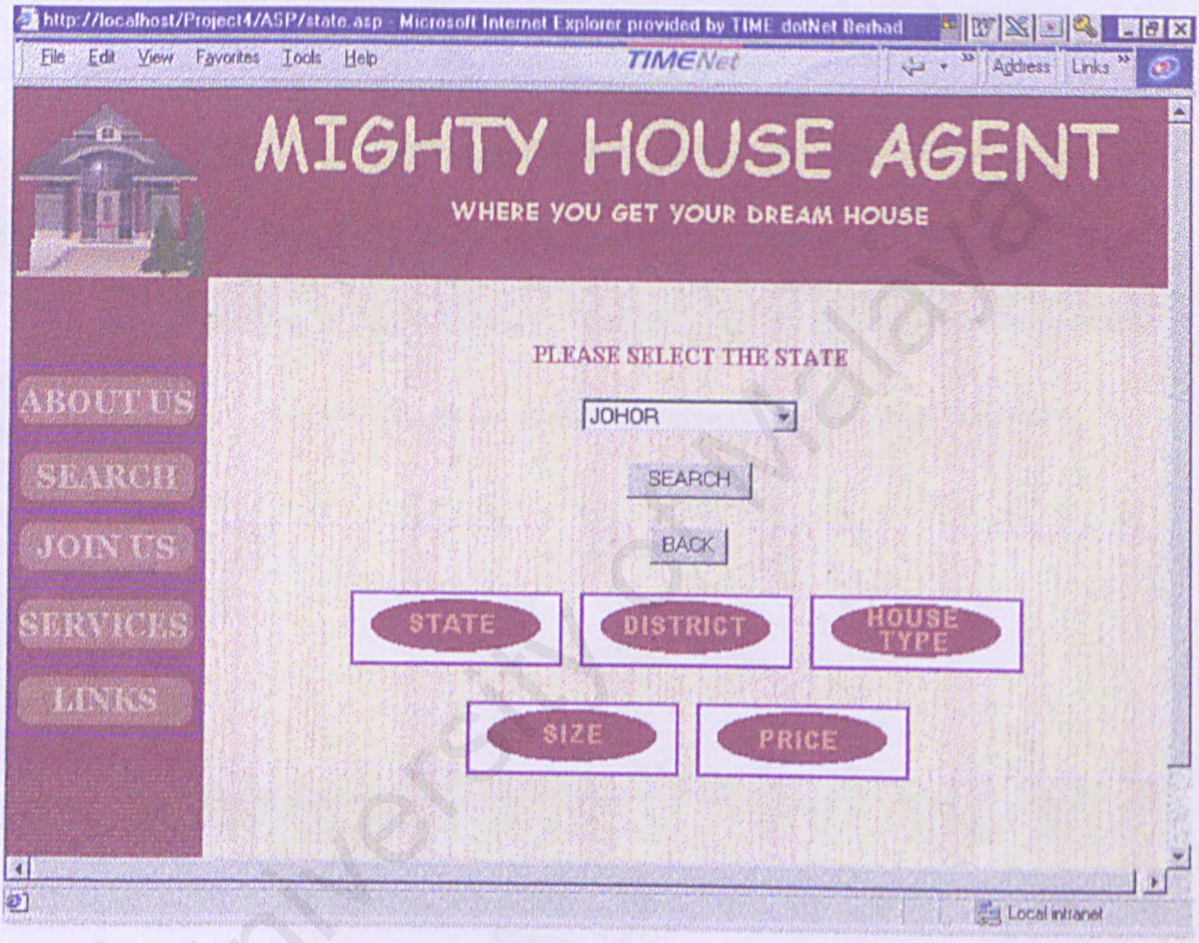
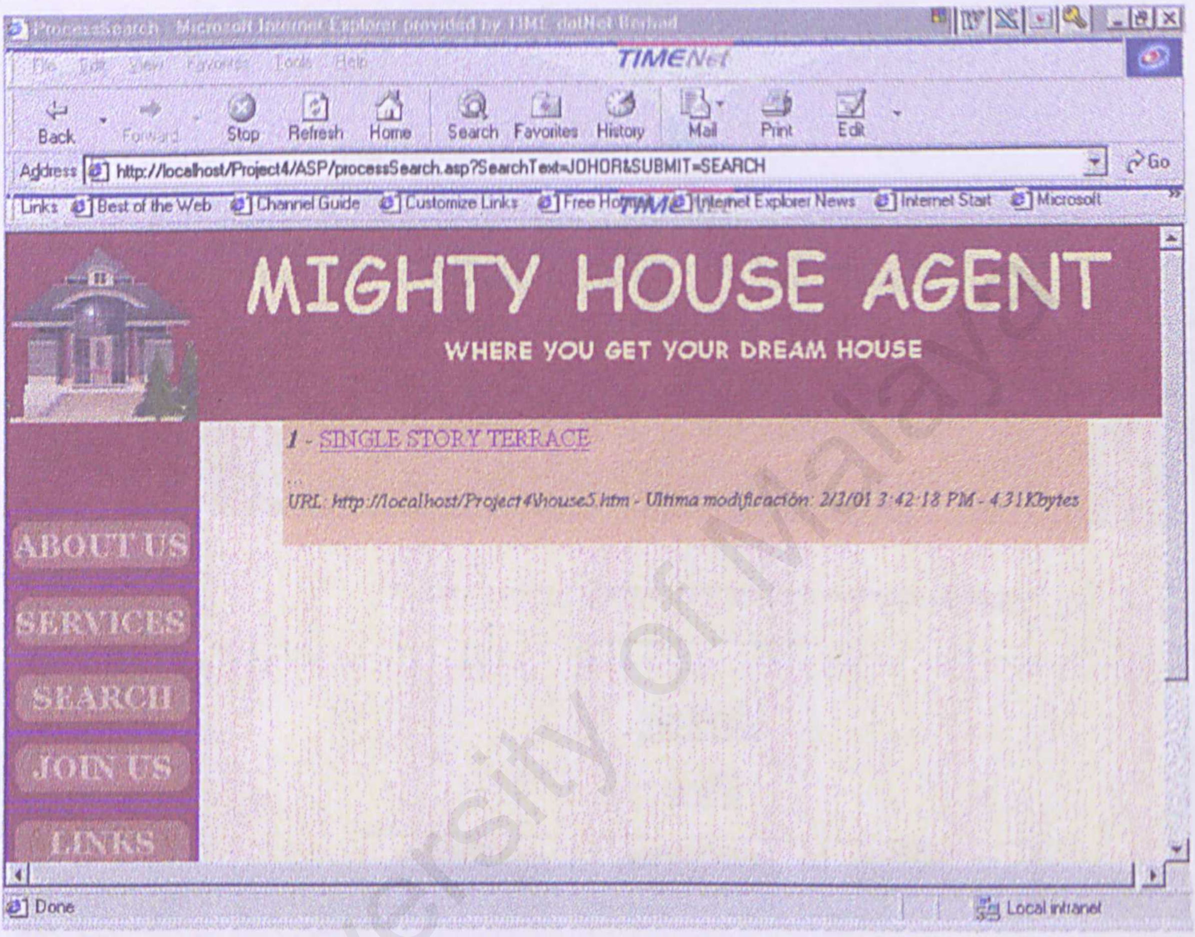

http://localhost/Project4/ASP/state.asp - Microsoft Internet Explorer provided by TIME dotNet Berhad
File Edit View Favorites Tools Help
TIMENet
Address Links
MIGHTY HOUSE AGENT
WHERE YOU GET YOUR DREAM HOUSE
ABOUT US
SEARCH
JOIN US
SERVICES
LINKS
PLEASE SELECT THE STATE
JOHOR
SEARCH
BACK
STATE
DISTRICT
HOUSE TYPE
SIZE
PRICE
Local intranet

Figure 2.4

The state is chosen from the list given in the listbox above the SEARCH button and the SEARCH button is clicked. The result of the request would be displayed. Please refer to Figure 2.5. All the search result page would be similar to Figure 2.5.



ProcessSearch - Microsoft Internet Explorer provided by TIME dotNet Berhad

Address: http://localhost/Project4/ASP/processSearch.asp?SearchText=JOHOR&SUBMIT=SEARCH

1 - SINGLE STORY TERRACE

...

URL: http://localhost/Project4/house5.htm - Ultima modificación: 2/3/01 3:42:18 PM - 4.31Kbytes

Figure 2.5

2.3.2 SEARCH BY DISTRICT/AREA

When the **DISTRICT** button from the search page is clicked a page similar to Figure 2.6 is displayed.

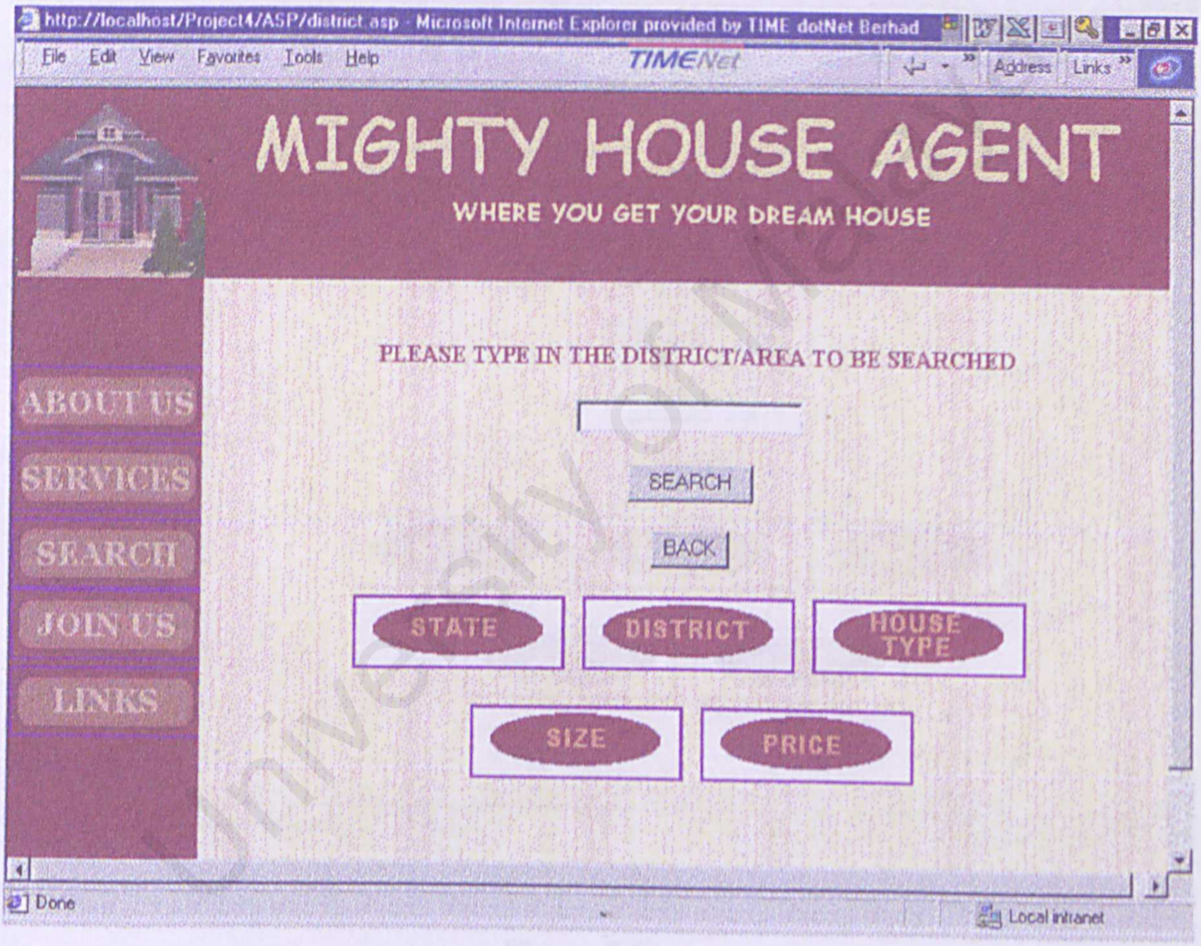

http://localhost/Project4/ASP/district.asp - Microsoft Internet Explorer provided by TIME dotNet Berhad
File Edit View Favorites Tools Help
TIMENet
Address Links
MIGHTY HOUSE AGENT
WHERE YOU GET YOUR DREAM HOUSE
ABOUT US
SERVICES
SEARCH
JOIN US
LINKS
PLEASE TYPE IN THE DISTRICT/AREA TO BE SEARCHED
SEARCH
BACK
STATE
DISTRICT
HOUSE TYPE
SIZE
PRICE
Done
Local intranet

Figure 2.6

The district or area name is typed into the empty space above the SEARCH button and the SEARCH button is clicked. The result of the request would be displayed.

2.3.3 SEARCH BY HOUSE TYPE

When the **HOUSE TYPE** button from the search page is clicked a page similar to Figure 2.7 is displayed.

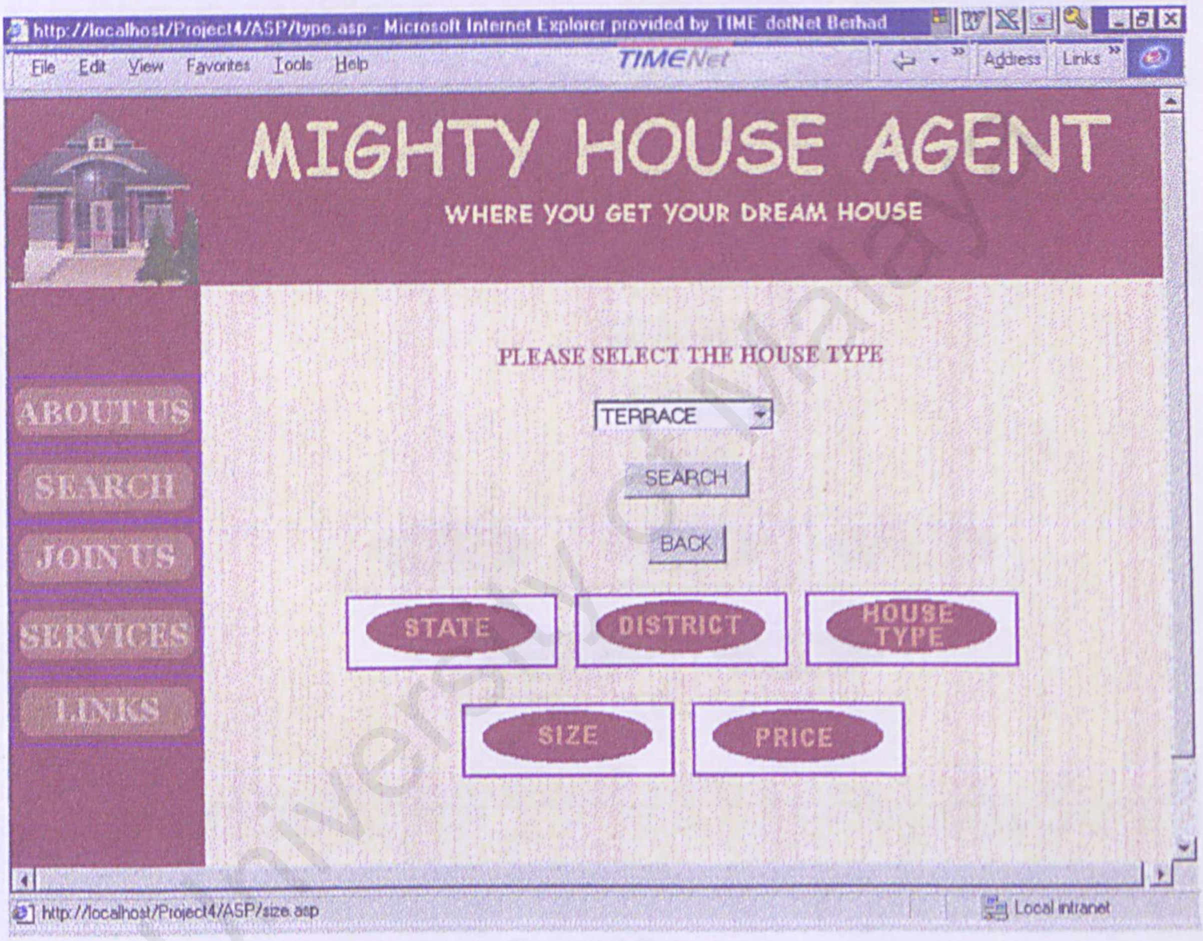
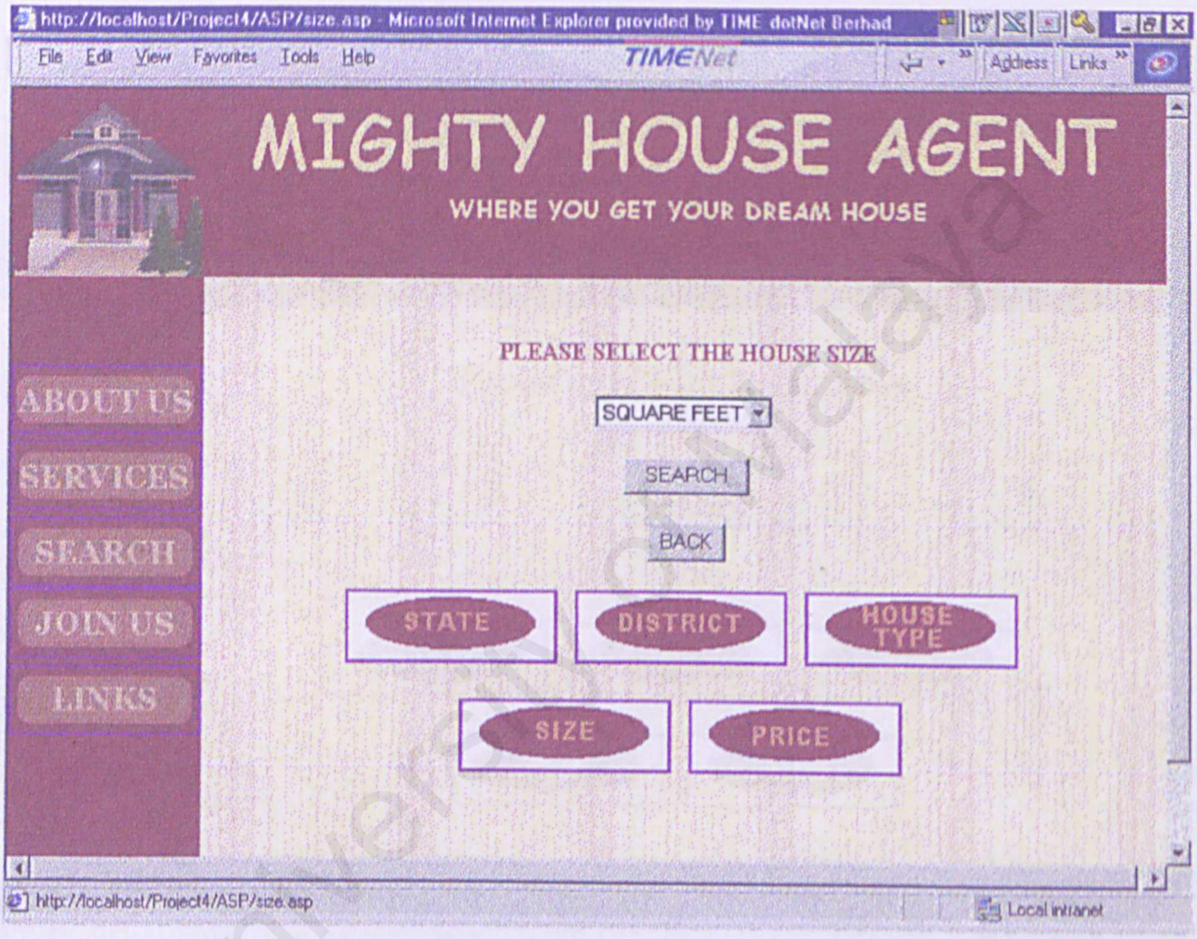

http://localhost/Project4/ASP/type.asp - Microsoft Internet Explorer provided by TIME dotNet Berhad
File Edit View Favorites Tools Help
TIMENet
Address Links
MIGHTY HOUSE AGENT
WHERE YOU GET YOUR DREAM HOUSE
ABOUT US
SEARCH
JOIN US
SERVICES
LINKS
PLEASE SELECT THE HOUSE TYPE
TERRACE
SEARCH
BACK
STATE
DISTRICT
HOUSE TYPE
SIZE
PRICE
http://localhost/Project4/ASP/size.asp
Local intranet

Figure 2.7

The house type is chosen from the list given in the listbox above the SEARCH button and the SEARCH button is clicked. The result of the request would be displayed. Please refer to Figure 2.5.

2.3.4 SEARCH BY HOUSE SIZE

When the **SIZE** button from the search page is clicked a page similar to Figure 2.8 is displayed.



http://localhost/Project4/ASP/size.asp - Microsoft Internet Explorer provided by TIME dotNet Berhad

MIGHTY HOUSE AGENT

WHERE YOU GET YOUR DREAM HOUSE

PLEASE SELECT THE HOUSE SIZE

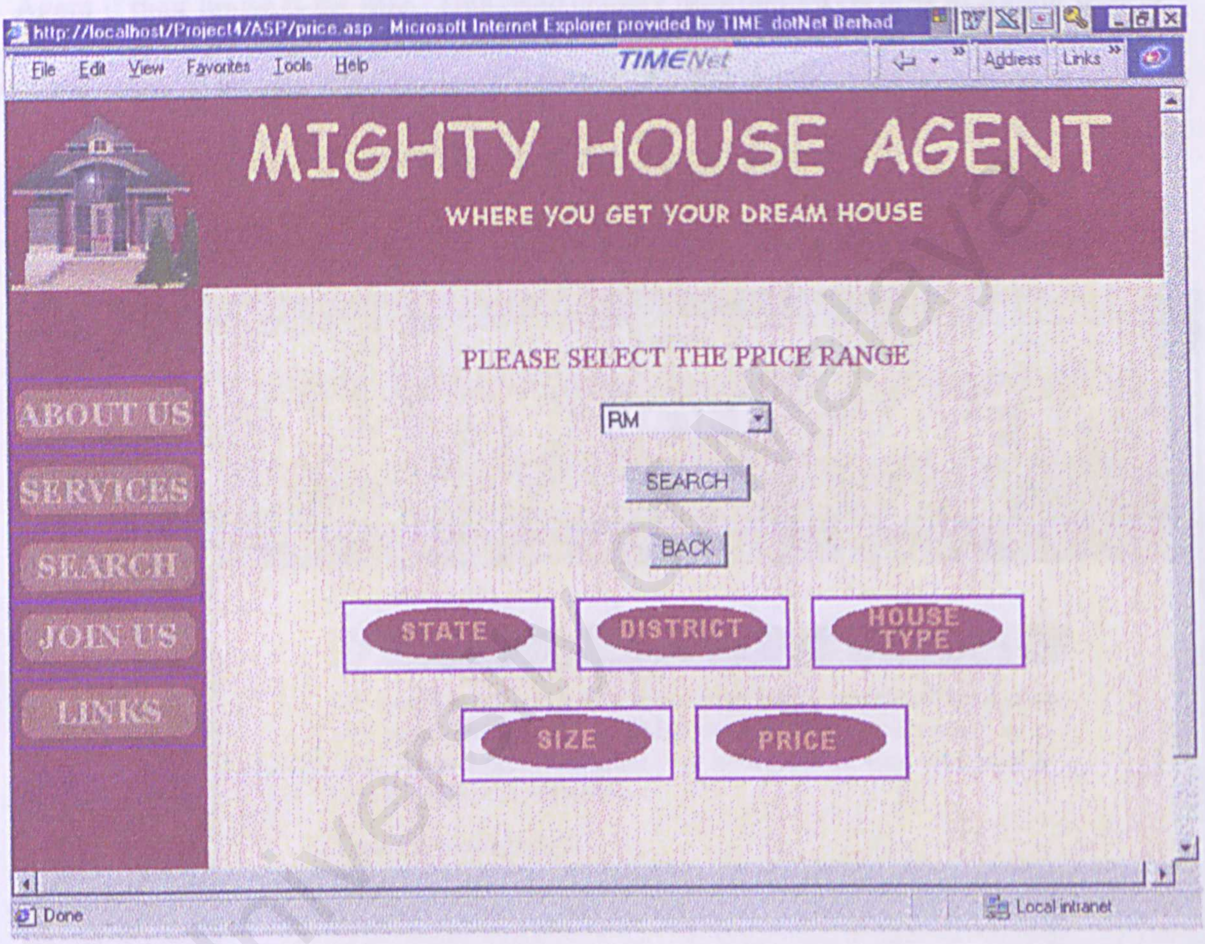
SQUARE FEET

Figure 2.8

The house size is chosen from the list given in the listbox above the SEARCH button and the SEARCH button is clicked. The size would include land as well if it is house with land. The size is given in a range so that it would be easier for the customer to select. The result of the request would be displayed. Please refer to Figure 2.5.

2.3.5 SEARCH BY HOUSE PRICE

When the **PRICE** button from the search page is clicked a page similar to Figure 2.9 is displayed.



PLEASE SELECT THE PRICE RANGE

RM

SEARCH

BACK

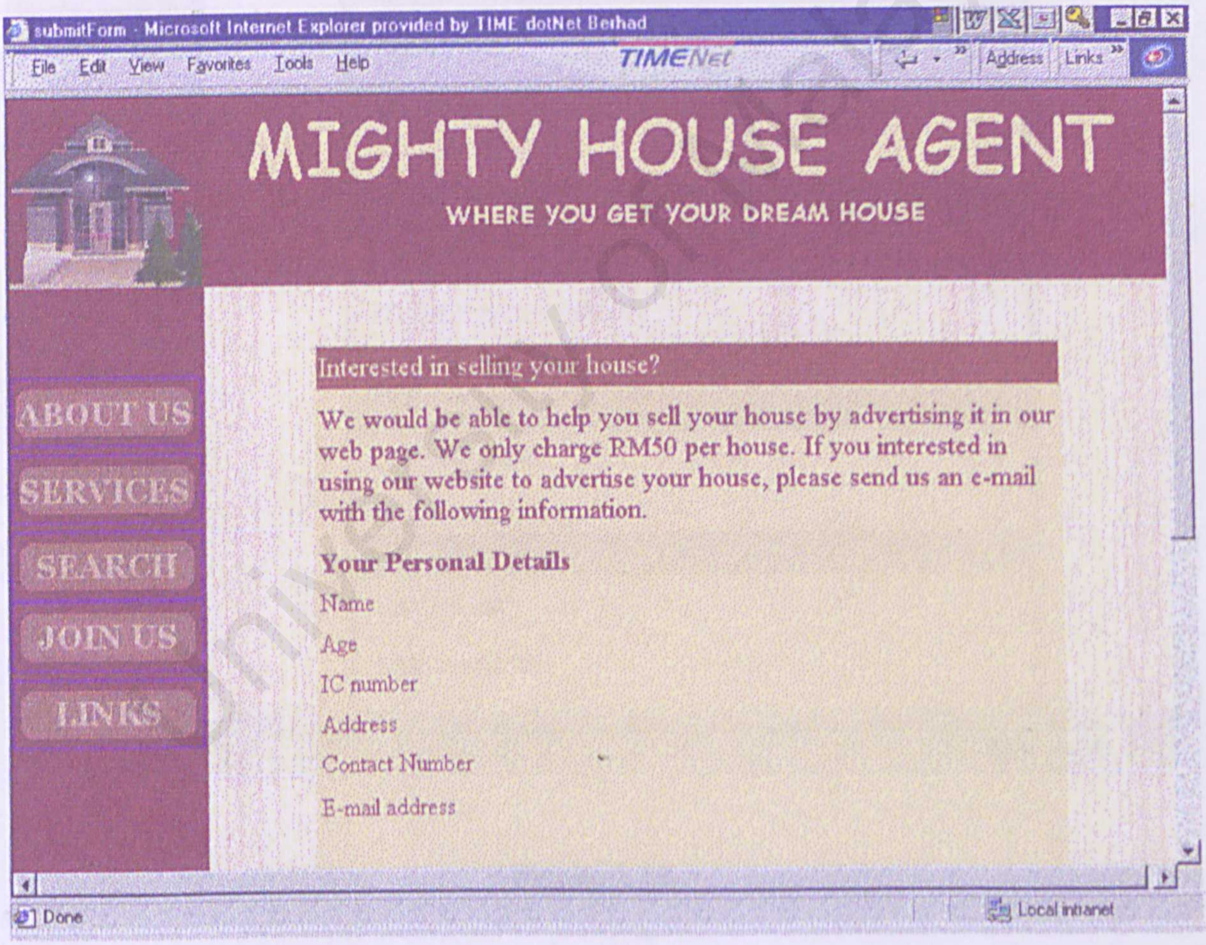
Figure.2.9

The house price is chosen from the list given in the listbox above the SEARCH button and the SEARCH button is clicked. The price is given in a range so that it would be easier for the customer to select. The result of the request would be displayed. Please refer to Figure 2.5.

2.4 TO ADVERTISE IN MIGHTY HOUSE AGENT

The **JOIN US** button by the side would display a page similar to Figure 2.10.

This page is meant for customers who are interested in advertising in Mighty House Agent if their house is for sale. This page display information on how to submit details of the house to be sold and the owner's details. All this details must be mailed to the given address of Mighty House Agent 's sales Administrator.


submitForm - Microsoft Internet Explorer provided by TIME dotNet Berhad
File Edit View Favorites Tools Help
TIMENet
Address Links
MIGHTY HOUSE AGENT
WHERE YOU GET YOUR DREAM HOUSE
ABOUT US
SERVICES
SEARCH
JOIN US
LINKS
Done
Local intranet

Interested in selling your house?

We would be able to help you sell your house by advertising it in our web page. We only charge RM50 per house. If you interested in using our website to advertise your house, please send us an e-mail with the following information.

Your Personal Details

Name

Age

IC number

Address

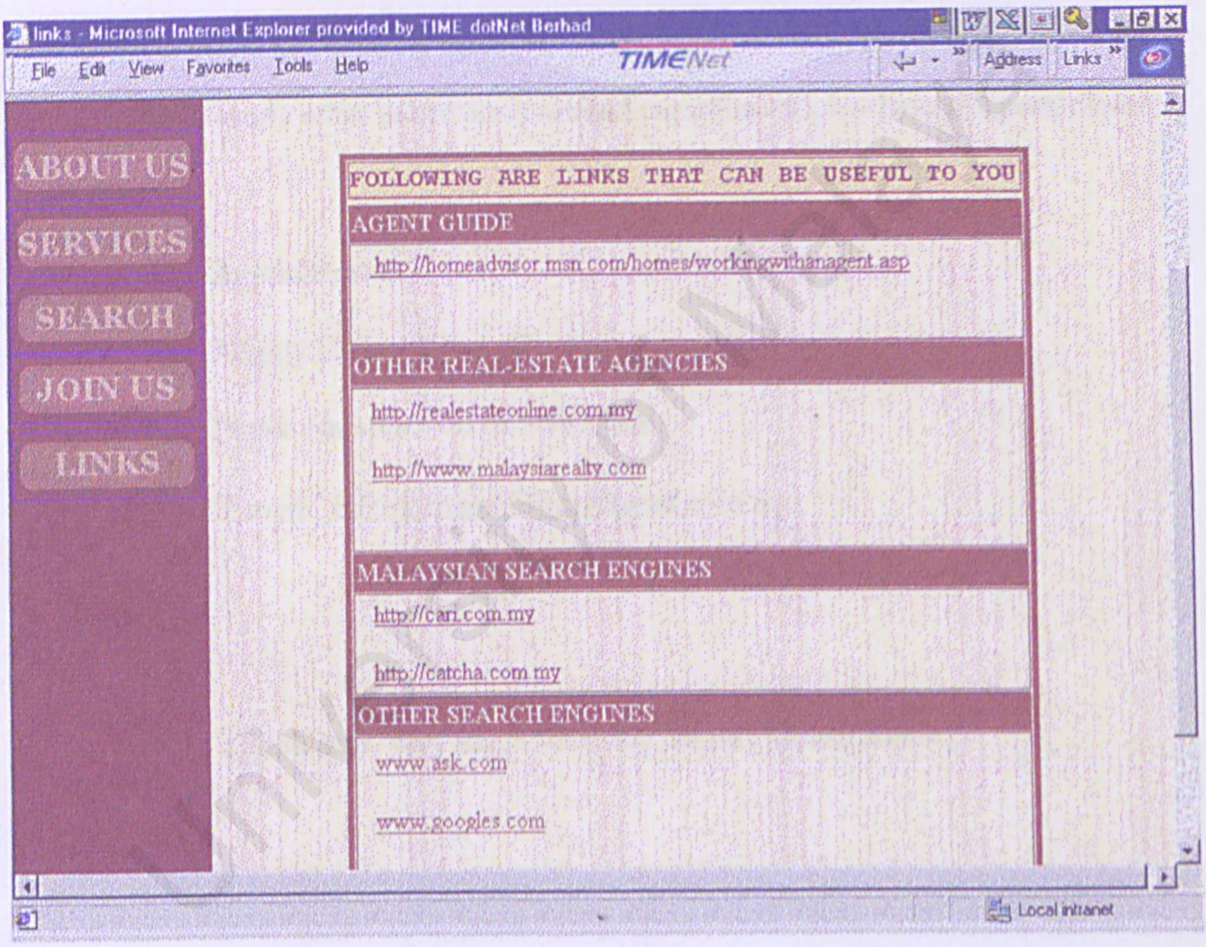
Contact Number

E-mail address

Figure 2.10

2.5 TO LINK TO OTHER WEB SITES

The **LINKS** button by the side would display the as the following(Figure 2.11). It shows some of the important links that may interest the customers.



FOLLOWING ARE LINKS THAT CAN BE USEFUL TO YOU

AGENT GUIDE

http://homeadvisor.msn.com/homes/workingwithanagent.asp

OTHER REAL-ESTATE AGENCIES

http://realestateonline.com.my

http://www.malaysiarealty.com

MALAYSIAN SEARCH ENGINES

http://cari.com.my

http://catcha.com.my

OTHER SEARCH ENGINES

www.ask.com

www.googles.com

Figure 2.11

CHAPTER 3 : CONCLUSION

Chapter 1 explains on the requirements for Mighty House Agent to be uploaded and also on how to get it working.

Chapter 2 of the user manual has described the ways to use Mighty House Agent and all of its services. Explanation has been given on functions of all icons and buttons to help users of the system to use Mighty House Agent without any trouble.

If there is any other information needed regarding Mighty House Agent, please contact:

Administrator,
Mighty House Agent Sdn Bhd.
Phone Number : +6 03 389 7339
E-mail : Info@MightyHouseAgent.com.my

Thank you very much for using Mighty House Agent.

SAMPLE SOURCE CODE OF MIGHTY HOUSE AGENT

SAMPLE SEARCH FORM : SEARCH BY STATE

```
<html>
<head>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</head><body bgcolor="[illegible]">
<div align="left" id="Layer1" style="BACKGROUND-COLOR: [illegible]; BORDER-
BOTTOM: #000000 1px; BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 124px; LEFT: 124px; POSITION:
absolute; TOP: 0px; WIDTH: [illegible]px; Z-INDEX: 1; layer-background-color:
#CC99FF">
   <p><center><b><font color="white" size="6" face="sans-serif" style="COLOR:
moccasin; FONT-FAMILY: cursive; FONT-SIZE: xx-large">MIGHTY HOUSE
AGENT</font></b></center>
<center>&nbsp;
   <b><font color="white" face="sans-serif" style="COLOR: moccasin; FONT-
FAMILY: cursive; [illegible]</font></b></center>
</div>
<div align="left" [illegible] BORDER-
BOTTOM: [illegible] #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 45px; LEFT: 0px; POSITION:
absolute; TOP: 125px; WIDTH: [illegible]; Z-INDEX: 2; layer-background-color:
#99CCFF">
<p>&nbsp;</p>
<p>&nbsp;<A href="../html/aboutus.html"><IMG height=42
src="../images/ABOUTUS.jpg" width=126></A>
<A href="query1.asp"><IMG height=42 src="../images/SEARCH1.jpg"
width=126></A>&nbsp;
<A href="../html/contact.html"><IMG height=42 src="../images/CONTACTS.jpg"
width=126></A>
<A href="../html/services.html"><IMG height=42 src="../images/SERVICES.jpg"
width=126>
<A href="../html/links.html"><IMG height=42 src="../images/LINKS.jpg"
width=126></A></p>
</div>
<div id="Layer3" style="BACKGROUND-COLOR: #ffffff; BACKGROUND-
IMAGE: url(../images/bg1.jpg); BORDER-BOTTOM: #000000 1px; BORDER-
LEFT: #000000 1px; BORDER-RIGHT: #000000 1px; BORDER-TOP: #000000 1px;
HEIGHT: [illegible]px; LEFT: 10px; POSITION: absolute; TOP: 170px; WIDTH: [illegible]px; Z-
INDEX: 3; layer-background-color: #FFCC99">
<div id="Layer4" style="BACKGROUND-COLOR: white; BACKGROUND-IMAGE:
url(../images/bg1.jpg); BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: [illegible]px; LEFT: [illegible]px; POSITION:
```

SAMPLE SOURCE CODE OF MIGHTY HOUSE AGENT

SAMPLE SEARCH FORM : SEARCH BY STATE

```
<html>
<head>
<meta NAME="GENERATOR" Content="Microsoft Visual Studio 6.0">
</head><body bgcolor="#ffffff">
<div align="left" id="Layer1" style="BACKGROUND-COLOR: maroon; BORDER-
BOTTOM: #000000 1px; BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 124px; LEFT: 124px; POSITION:
absolute; TOP: 0px; WIDTH: 660px; Z-INDEX: 1; layer-background-color:
#CCFFFF">
    <p><center><b><font color="mistyrose" size="6" face="sans-serif" style="COLOR:
moccasin; FONT-FAMILY: cursive; FONT-SIZE: xx-large">MIGHTY HOUSE
AGENT</font></b></center>
<center>&nbsp;
    <b><font color="white" face="sans-serif" style="COLOR: moccasin; FONT-
FAMILY: cursive">WHERE YOU GET YOUR DREAM HOUSE</font></b>
</div>
<div align="left" id="Layer2" style="BACKGROUND-COLOR: maroon; BORDER-
BOTTOM: #000000 1px; BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 438px; LEFT: 0px; POSITION:
absolute; TOP: 124px; WIDTH: 133px; Z-INDEX: 2; layer-background-color:
#99CCFF">
<p>&nbsp;</p>
<p>&nbsp;<A href="../html/aboutUs.html"><IMG height=42
src="../images/ABOUTUS.jpg" width=126></A>
<A href="queryForm.asp"><IMG height=42 src="../images/SEARCH.jpg"
width=126></A>&nbsp;
<A href="../html/submit.html"><IMG height=42 src="../images/JOINUS.jpg"
width=126></A>
<A href="../html/services.html"><IMG height=42 src="../images/SERVICES.jpg"
width=126></A>
<A href="../html/links.html"><IMG height=42 src="../images/LINKS.jpg"
width=126></A> </p>
</div>
<div id="Layer3" style="BACKGROUND-COLOR: #ffffff; BACKGROUND-
IMAGE: url(../images/h2n1.jpg); BORDER-BOTTOM: #000000 1px; BORDER-
LEFT: #000000 1px; BORDER-RIGHT: #000000 1px; BORDER-TOP: #000000 1px;
HEIGHT: 134px; LEFT: -10px; POSITION: absolute; TOP: -10px; WIDTH: 139px; Z-
INDEX: 3; layer-background-color: #FFFFFF"></div>
<div id="Layer4" style="BACKGROUND-COLOR: white; BACKGROUND-IMAGE:
url(../images/bac1.1.jpg); BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 438px; LEFT: 132px; POSITION:
```

SAMPLE SOURCE CODE OF MIGHTY HOUSE AGENT

```
absolute; TOP: 124px; WIDTH: 653px; Z-INDEX: 4;   layer-background-image:
url(Visual%20Studio%20Projects/Project4/Project4_Local/ASP/welcomePage.asp)">
<p align="center"><strong>
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;</strong></p>
<p align="center"><strong style="COLOR: darkred">PLEASE SELECT THE
STATE</strong>
</p>
<p align="center">
          <form action="processSearch.asp" method="get" name="searchForm">
     <select id="SearchText" name="SearchText" TYPE="HIDDEN">
          <optionselected>JOHOR</option><option>KEDAH</option>
      <option>KELANTAN</option><option>KUALA LUMPUR</option>
      <option>MELAKA</option><option>NEGERI SEMBILAN</option>
      <option>PAHANG</option><option>PERAK</option>
      <option>PERLIS</option><option>PULAU PINANG</option>
      <option>SABAH</option><option>SARAWAK</option>
      <option>SELANGOR</option><option>TERENGGANU</option>
                                        </select>
          <p><input type="submit" value="SEARCH" id="SUBMIT"
name="SUBMIT"></p>
          </form> <INPUT id=btnBack name=btnBack
onclick=Javascript:history.back(1)  type=button value=BACK>
<P></P>
<P>
             <A href="state.asp"><IMG height=44 src="../images/STATE.gif"
width=139></A>&nbsp;&nbsp;
                                                  <A href="district.asp"><IMG height=44
src="../images/DISTRICT.gif" width=139></A>&nbsp;&nbsp;
                                                  <A href="type.asp"><IMG height=44
src="../images/TYPE.gif" width=139></A>&nbsp;</P>
<P>&nbsp;
             <A href="size.asp"><IMG height=44 src="../images/SIZE.gif"
width=139></A>&nbsp;&nbsp;
                                     <A href="price.asp"><IMG height=44
src="../images/PRICE.gif" width=139></A>&nbsp;&nbsp; </P>
                              <p></p>
</div></CENTER>
</body>
</html>
```

SAMPLE SOURCE CODE OF MIGHTY HOUSE AGENT

PROCESS SEARCH

```
<%@ Language=VBScript %>
<html>
<head>
<title>ProcessSearch</title>
</head>
<body bgcolor="#ffffff">
<div align="left" id="Layer1" style="BACKGROUND-COLOR: maroon; BORDER-
BOTTOM: #000000 1px; BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 124px; LEFT: 124px; POSITION:
absolute; TOP: 0px; WIDTH: 660px; Z-INDEX: 1; layer-background-color:
#CCFFFF">
   <p><center><b><font color="mistyrose" size="6" face="sans-serif" style="COLOR:
moccasin; FONT-FAMILY: cursive; FONT-SIZE: xx-large">MIGHTY HOUSE
AGENT</font></b></center>
<center>&nbsp;
   <b><font color="white" face="sans-serif" style="COLOR: moccasin; FONT-
FAMILY: cursive">WHERE YOU GET YOUR DREAM HOUSE</font></b>
</div>
<div align="left" id="Layer2" style="BACKGROUND-COLOR: maroon; BORDER-
BOTTOM: #000000 1px; BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 438px; LEFT: 0px; POSITION:
absolute; TOP: 124px; WIDTH: 125px; Z-INDEX: 2; layer-background-color:
#99CCFF">
<p>&nbsp;</p>
<p>&nbsp;<A href="../html/aboutUs.html"><IMG height=42
src="../images/ABOUTUS.jpg" width=126></A>
<A href="../html/services.html"><IMG height=42 src="../images/SERVICES.jpg"
width=126></A>
<A href="queryForm.asp"><IMG height=42 src="../images/SEARCH.jpg"
width=126></A>&nbsp;
<A href="../html/submit.html"><IMG height=42 src="../images/JOINUS.jpg"
width=126></A>
<A href="../html/links.html"><IMG height=42 src="../images/LINKS.jpg"
width=126></A> </p>
</div>
<div id="Layer3" style="BACKGROUND-COLOR: #ffffff; BACKGROUND-
IMAGE: url(../images/h2n1.jpg); BORDER-BOTTOM: #000000 1px; BORDER-
LEFT: #000000 1px; BORDER-RIGHT: #000000 1px; BORDER-TOP: #000000 1px;
```

SAMPLE SOURCE CODE OF MIGHTY HOUSE AGENT

```
HEIGHT: 134px; LEFT: -10px; POSITION: absolute; TOP: -10px; WIDTH: 139px; Z-
INDEX: 3; layer-background-color: #FFFFFF"></div>
<div id="Layer4" style="BACKGROUND-COLOR: white; BACKGROUND-IMAGE:
url(../images/bac1.1.jpg); BORDER-LEFT: #000000 1px; BORDER-RIGHT: #000000
1px; BORDER-TOP: #000000 1px; HEIGHT: 438px; LEFT: 132px; POSITION:
absolute; TOP: 124px; WIDTH: 653px; Z-INDEX: 4;   layer-background-image:
url(Visual%20Studio%20Projects/Project4/Project4_Local/ASP/welcomePage.asp)"
align="center">

<p><% if Request("SearchText") <> "" then %>

  <p><%
   Const fsoForReading = 1
   Dim objFile, objFolder, objSubFolder, objTextStream
   Dim bolCase, bolFileFound
   dim strDeTag, Ext, strFile, strContent, strRoot, strTag, strText, strText1, strTitle,
strTitleL
   Dim reqLength, count
   strFile = " .htm "
   strRoot = "/"
   strText = Request("SearchText")

   Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
   Set objFolder = objFSO.GetFolder(Server.MapPath(strRoot))
   CurURL= "http://" & Request.serverVariables("SERVER_NAME")
   CurPath = objFolder
   schSubFol(objFolder)
   For Each objSubFolder in objFolder.SubFolders
         schSubFol(objSubFolder)
   Next
   If Not bolFileFound then Response.Write "<font size=6 type=arial
 color=darkred><b>Sorry!!!<p>There is no house which meets this
 requirement.</b></font></p><p><INPUT id=btnBack name=btnBack
 onclick=Javascript:history.back(1)  type=button value=Back></p>"
    Set objTextStream = Nothing
    Set objFolder = Nothing
    Set objFSO = Nothing
    Sub schSubFol(objFolder)
         For Each objFile in objFolder.Files
                  If Count + 1 > strText = "" Then Exit Sub
                  If Response.IsClientConnected Then
                           strext = right(objFile.Name,3)
                           if instr(1,strFile,strext) > 0 then
```

SAMPLE SOURCE CODE OF MIGHTY HOUSE AGENT

```
                              Set objTextStream =
objFSO.OpenTextFile(objFile.Path,fsoForReading)
                              strContent = objTextStream.ReadAll
                              If InStr(1,strContent,strtext) > 0 Then
                              postitle = InStr(1, strContent, "<TITLE>",1)
                              If postitle > 0 Then
                                      strTitle = Mid(strContent, postitle + 7, InStr(1,
strContent, "</TITLE>", 1) - (postitle + 7))
                              Else
                                      strTitle = "search result"
                              end if
                              Count = Count + 1
                              Response.Write "<p><table
bgcolor=tan><tr><td><DL><DT><B><I>"& Count &"</I></B> - <A
HREF="&Obt_Url(objFile.path) & ">" & strTitle &
"</A></DT><BR><DD></td></tr><tr><td>"
                              strTitleL = InStr(1, strContent, "</TITLE>", 1) - InStr(1,
strContent, "<TITLE>", 1) + 7
                              strDeTag = ""
                              bolTagFound = False

                              Do While InStr(strContent, "<")
                                      bolTagFound = True
                                      strDeTag = strDeTag & " " & Left(strContent,
InStr(strContent, "<") - 1)
                                      strContent = MID(strContent, InStr(strContent,
">") + 1)
                              Loop
                              strDeTag = strDeTag & strContent
                              If Not bolTagFound Then strDeTag = strContent
                                      If reqLength = "0" Then
                                              Response.Write obt_url(objFile.Path) &
"</DD></DL></td></tr></table>"
                                      Else
                                              Response.Write Mid(strDeTag, strTitleL,
reqLength) & "...<BR><I><FONT SIZE='2'>URL: " & obt_url(objFile.Path) & " -
Ultima modificación: " & objFile.DateLastModified & " - " &
FormatNumber(objFile.Size / 1024) &
"Kbytes</FONT></I></DD></DL></td></tr></table>"
                                      end if
                                      bolFileFound = True
                              End If
                              objTextStream.Close
```

```
                        End If
                  End If
         Next
    End Sub
    Function Obt_Url (nompath)
         obt_url = CurUrl +"/"+ right(nompath,len(nompath) - len(curpath)-1)
    end function
    %>
    <% end if %></p>
<p></p>

   </div></CENTER>
</body>
</html>
```