NAME: **NORMALYNA ABDUL MALEK**

MATRIC NO.: **WET020119**

SUBJECT CODE: **WXES3181/2**

PROJECT TITLE:

IMAGE RETRIEVAL SYSTEM FOR WEB

SUPERVISOR NAME:

MDM. NORJIHAN ABDUL GHANI

MODERATOR NAME:

ASSOC. PROF. DR. DILJIT SINGH BALWANT SINGH

ABSTRACT

The project title is Image Retrieval System for Web. This project is developed for the subject 'Projek Ilmiah Tahap Akhir'. The system is a web based system. It is for the user to retrieve similar shape of images. The user for this system is any users who have internet connection and the administrator. The overall project covers the Introduction, Literature review, Methodology, System Analysis, System Design, System Implementation, System Testing and System Evaluation. This system is divided into two modules that will be developed by two people. Module 1 is to do image analysis and convert it into image data while Module 2 covers the matching of image query from the system and the development of interfaces for the overall system. The research made was from internet resources, article or journal and also discussion with the supervisor and the moderator. The techniques used for this system is the Sobel Edge Detector, Reduced Chain Code, User Interface Design technique and the Weighted Levenshtein Distance. The system life-cycle model used is the Iteration-and-Incremental Model. The main aim for this project is to achieve all of the Objectives at the end of the Image Retrieval System for Web development.

ACKNOWLEDGEMENT AND DEDICATION

I'm grateful to Allah that I have finished the first part of this project. With all the blessed I'm hoping that this project will be a successful project.

Thank you to my family especially to my dad, Abdul Malek Chua and mum, Flora Harryissu@Norhidayati who is always there to support me in everything I do. Without them I would never gone this far.

I also want to thank to our supervisor, Madam Norjihan Abdul Ghani. Thank you for being a supportive and always give us guidance any time we needed it.

To our moderator, Assoc. Prof. Dr. Diljit Singh, I want to say thank you for all the comments and suggestions that makes our project a better project.

To my partner in this system development, Suhaili Beeran Kutty, thank you so much for being such a good friend and also a great partner in developing this system.

I'm also thankful to my friend in KUKUM, all my friends at the FSKTM and housemates because they always give me all the strength I need to be able to finish this project.

Finally, a thousand thank you for all the people that are involved in this project.

TABLE OF CONTENTS

LIST OF TABLES **PAGE**

LIST OF FIGURES **PAGE**

CHAPTER 1:

INTRODUCTION

CHAPTER 1: Introduction

1.1 Introduction

The system that will be developed is The Image Retrieval System for Web. It is an online system that characterizes and retrieves images based on similarity of image content.

Content based similarity retrieval can be using visual texture that is color, texture, spatial localization and shape (Marinette, Ali & Henri, 1999). This system uses shape similarity based retrieval because among this entire visual feature, shape is the most important feature. It is important because shape feature represents significant regions or relevant objects in images. The Image Retrieval System will be divided into two modules:

Module 1:

The extraction of shape features from an input image and the image data representation. First is the extraction process. It extracts shape features that are the edges from an image by using detection techniques. Then the extracted feature is associated with image data representations stored in the database.

Module 2:

In the second module, interface is developed for users to formulate their query and matching technique is implemented for image retrieval. The user query contains image data generated by the system. Then the retrieval process begins to computes the similarity between user query and database image data and sorts the best similar images according to their similarity value. The outputs retrieved are images that have similar shapes in the interface.

Both of these modules complement each other in the development of the system.

1.2 Objectives

The objectives of the Image Retrieval System for Web are:

- To characterize and retrieve images based on their shape
- To make it easier for the user to search for similar images in the database using input image or menu-driven query
- To analyze and code the images using Sobel, Reduced Chain Code and weighted Levensthein distance
- To provide a user friendly interface for this web-based system for the ease of use
- To combine both of the modules developed during the implementation of the system.

1.3 Scope

The user of the Image Retrieval System for Web is the public and the administrator of the system. The system covers only for the small application use such as using the images retrieved as images for teaching student the basic shapes of images or the images used in developing other interfaces in other web-based system.

The public

The public can be anyone who has access to the internet. They can use the system if their looking for images according to the shape. They could retrieve the images by using menu of shapes provided in the interface of the system or by input image to the system. The output is images that is similar to the shape chosen or image input by the user. This type of user will be novice user who is a less experienced computer user and does not need any training to use the system.

The administrator

The administrator is the one who develop the system. The administrator will upload all the original images into the image database and maintain the system from time to time.

1.4 Overview of the system

As stated in the introduction, the Image Retrieval System for Web is a web-based system developed for the user to retrieve images online based on the similarity of the image shapes.

The two modules in this system are the extraction of visual features from image to represent it in image data and the developing the user interface for querying the image.

In the Module 1, there are 2 methods of analysis technique used for the extraction of visual features. The analysis technique involved is edge detection and chain coding (outline based features). The codes use to represent the image is generated from the image analysis will be kept in the image database.

The edge detection technique that is used is the Sobel Edge Detector. Sobel Edge detector uses a simple convolution kernel to create a series of gradient magnitudes (*Internet Reference, 12/07/2004)*. There are two convolution kernels in Sobel, one is to detect changes in vertical contrast and the other is to detect horizontal contrast.

Then, the chain code is used for representing the boundary of any shape. But the basic chain code invariant to scale and rotation, so the modified chain code is used. The type of modified chain code for this system is the Reduced Chain Code (RCC). The RCC is the representation of shape as image data in the database.

Module 2 for the image retrieval of this system is developing a user interface so the user can formulate their query to retrieve images. The query can be made in the interface by using two ways;

i) Menu-based

- Basic shapes are provided for the user to choose from basic shape such as circle, rectangle, square and polygon

ii) Input image

- The second one is where the user can input an image to search for the similar images from the database.

Both of the queries will go through the analysis technique and after that then the process of computing the similarity of the image input with the images in the database will be done.

The Weighted Levensthein Distance is used to compute the similarity between source and target features and sorts the best similar images according to their similarity value. This is the process of image matching to search for the queried images.

The retrieval process can be express in this following retrieval model:

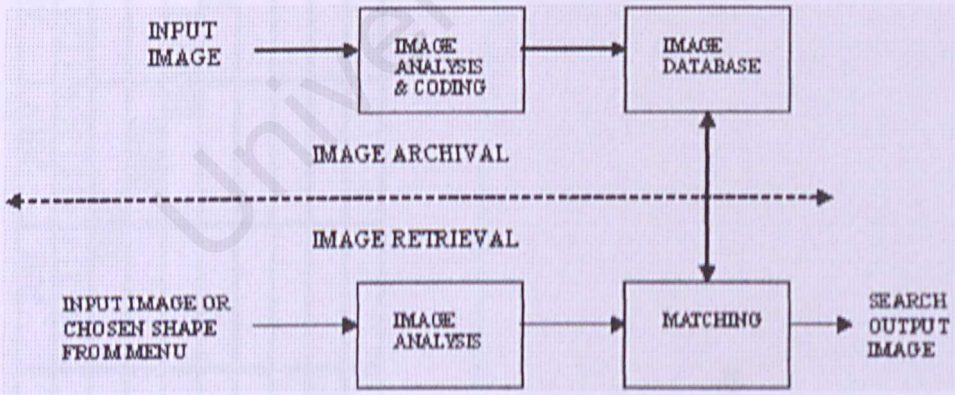


Figure 1.1: The retrieval model

1.5 Project Schedule

	JUN				JULY				AUG				SEPT				OCT				NOV				DEC				JAN				FEB				
Task	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	**Duration**
Research on Thesis Title																																					2 weeks
Requirement Analysis																																					2 weeks
Literature Review																																					3 weeks
System Analysis																																					3 weeks
System Design																																					4 weeks
System Development																																					10 weeks
Implementation and Testing																																					5 weeks
Documentation																																					5 weeks

Task

Figure 1.2: Project Schedule

1.6 Chapter Summary

This chapter includes the Introduction, Objectives of the System, Scope, Overview of the System and The Project Schedule.

The introduction is general information of what to be developed that is the Image Retrieval System for Web. It explains about what techniques to be used in both module of the system. The objectives are for the developer to achieve throughout the system development. The scope for this system is for the administrator and the user. The overview of the system specifically explains about the system and the techniques used to build both of the modules in the system. The project schedule is the estimation of time to develop the entire system.

CHAPTER 2:

LITERATURE REVIEW

CHAPTER 2: Literature review

2.1 Research Method

i) Internet

The internet is where I get almost all the information that I need. I used the search engine such as Yahoo! and Google. It helps me in retrieving online researches, related article, journal, images for the database and other information.

ii) Document Room / Library

The thesis done by other people is used as my reference. It contains much information and guided me in writing my own thesis.

iii) Article or journal

Article or journal is one of the most important reference and source for my thesis. It is in a printed form but it is not so different from the online article or journal that I get from the internet.

iv) Discussion with project supervisor and moderator

I also have a discussion with our project supervisor and moderator. The discussion is mainly about the problems I face during the thesis writing and about the system to be developed.

2.2 Introduction of the system

The Image Retrieval System for Web to be developed is a web-based system. The reason why web-based system is developed is because the web applications had become the most popular applications in the world. Today's information is easily shared across computer networks in which some computers, called servers, offer common stores of programs and data that may be used by client computer distributed throughout the network.

By having an Internet connection, we can access to most of the Web applications available in the Internet. The World Wide Web (WWW) is a system of Internet servers that supports hypertext to access several Internet protocols on a single interface. It has its own protocol: Hypertext Transfer Protocol, or HTTP. It also applies to our system that will be uploaded into the World Wide Web.

The main task of the system is online image retrieval. The system retrieves images that have similar shapes. The user can query the image using menu provided or by input image to the system.

The process of image retrieval is divided into two modules. In the first module, the image input to the system. After that, the Sobel Edge Detector is use to detect the image edges. The boundary of shape will be change into image data and represented using Reduced Chain Code.

Module 2 of the system is where the users query for the images retrieval through the user interface of the system. The user interface is where the users can query the images from the system. To match the query with the images in the image database, the string measure distance technique, the Weighted Levensthein Distance is used. Then the

output images will be displayed. The original image, coding of the image and other attributes such as names, file type, description are kept in the database.

The overall technique of analyzing, coding, user interface, matching technique and all subject related to the system will be explained in the following literature review.

2.3 Tools and related technologies to the web application

2.3.1 Internet

For a web based system, connection to the internet is needed to publish and access the system.

The internet can be link by many different ways. As for a home computer, it may be linked to the Internet using a phone-line modem, DSL or cable modem that talks to an Internet service provider (ISP). A computer in a business or university will usually have a network interface card (NIC) that directly connects it to a local area network (LAN) inside the business.

Backbones around the world are connected through fiber-optic lines, undersea cables or satellite links. In this way, every computer on the Internet is connected to every other computer on the Internet.

2.3.2 Web server

A web server is a program that serves web pages upon request. Every Web server has an IP address and possibly a domain name. For example, if the user enters the URL http://www.geocities.com/home.html in your browser, this sends a request to the server

whose domain name is geocities.com. The server then fetches the page named home.html and sends it to the user's browser.

Web servers and browsers communicate using HTTP protocol. They come in various shapes and sizes and run under a variety of operating systems. IIS is the best Web server available for Windows 2000 and Windows XP.

2.3.3 Web Browser

Web browser is a computer program that implements the World Wide Web dokumen from all the wen browsers around the world. The web browser will use certain type of programming language to code and display the contents of a web site.

Communication between the web server and web browser is done using 'hypertext transfer protocol', also known as the HTTP. It is a protocol used to transfer files such as graphics and text. Examples of protocols are FTP, Gopher, and HTTPS. Web browser also access documents that is kept in other file format using the protocols.

Examples of web browser are as follows:

- Internet Explorer
- Mozilla
- Netscape Communicator

Web browser that will be used in this system is the Internet Explorer.

2.3.4 Tools

There are a variety of tools or software that is available to create hyper-text documents.

- Notepad
- Microsoft Front Page
- Microsoft Visual InterDev 6.0
- Macromedia Dreamweaver
- ASP.NET Web Matrix
- Others

By using the tools, the Web documents have to be saved as .htm or .html. Example: index.htm, main.html, test.aspx. The tool that will be used for the system is the Macromedia Dreamweaver. This is because it uses one integrated development environment to develop HTML, XHTML, XML, ASP, ASP.NET, JSP, PHP, and Macromedia ColdFusion websites. Another important tool is ASP.Net Web Matrix.

2.3.5 Programming Language

There are a variety of programming languages that are available to develop Web applications. Some examples are VBScript, JavaScript, ASP, ASP.NET, PHP, PERL, JSP and CGI.

ASP.NET is a unified Web development platform that provides the services necessary for developers to build enterprise-class Web applications. This is why it is chosen as the programming language for the system to be developed.

2.3.5.1 ASP.NET

ASP.NET is a compiled, .NET-based environment; it can be used to author applications in any .NET compatible language, including VB.NET, C#, and J#. Web Forms or XML Web services can be used when creating an ASP.NET application

ASP.NET Web Matrix is a freeware that used to build Web applications. Besides, Visual Studio.NET is a complete suite of tools for building both desktop and team-based Enterprise Web applications.

2.3.6 The database

2.3.6.1 SQL Servers

The Microsoft SQL Server is used as the database in this system. Its data is stored in databases. The data in a database is organized into the logical components visible to users. A database is also physically implemented as two or more files on disk.

Table 2.1 Table properties in the SQL Server

Data Type	Description	Default value
Column Name	The name of a column in a table. Column names must conform to rules for identifiers and must be unique in the table.	Blank
Datatype	The data type of the column. System- or user-defined data types are acceptable.	Character (char)
Length	The maximum number of digits (for numeric data types) or characters allowed for values in the column.	Differs for different data types (e.g., 10 for Character, 50 for VARBINARY)
Precision	The maximum total number of decimal digits that can be stored, both to the left and to the right of the decimal point.	0
Scale	The maximum number of decimal digits that can be stored to the right of the decimal point. This value must be less than or equal to the precision. Applies only to DECIMAL and NUMERIC data types.	0
Allow Nulls	Whether or not the column can accept null values.	Yes (selected)
Default Value	The value that will be inserted into the column if the user does not make an entry. Default values are ignored for columns with a timestamp data type. If you do not define a default value and a column allows nulls, NULL will be inserted.	Blank
Identity	Whether or not the column will generate incremental values for new rows based on the **Identity Seed** and **Identity Increment** settings.	No (not selected)
Identity Seed	The value assigned to the first row in the table. If the **Identity** setting is No, **Identity Seed** is blank. If the **Identity** setting isYes, **Identity Seed** defaults to 1.	Blank or 1.
Identity Increment	The value which is added to the **Identity Seed** and assigned to the second row in the table. Each subsequent row is increased by this value. If the **Identity** setting is No, **Identity Increment** is blank. If the **Identity** setting is Yes, **Identity Increment** defaults to 1.	Blank or 1.

Table 2.2 Data types in the SQL Server table

Logical Data Type to be stored in field)	**Physical Data Type Microsoft *SQLServer***
Fixed length character data *(use for fields with relatively fixed length character data)*	CHAR (size) *or* character (size)
Variable length character data *(use for fields that require character data but for which size varies greatly--such as ADDRESS)*	VARCHAR (max size) *or* character varying (max size)
Very long character data *(use for long descriptions and notes--usually no more than one such field per record)*	TEXT
Integer number	INT (size) *or* integer *or* smallinteger *or* tinuinteger
Decimal number	DECIMAL (size, decimal places) *or* NUMERIC (size, decimal places)
Financial Number	MONEY
Date (with time)	DATETIME or SMALLDATETIME Depending on precision needed
Current time (use to store the data and time from the computer's system clock)	TIMESTAMP
Yes or No; or True or False	BIT
Image	IMAGE
Hyperlink	VARBINARY
Can designer define new data types?	YES

2.4 Image and image type

Image is composed of pixels or picture elements. Picture resolution is the number of pixels or sample used to represent the image. Images are normally converted to file for storage and transfer. The file type is a special format for ordering and storing the bytes that make up the image. File types or formats are not necessarily compatible It is a must to match the file type with the application. Examples of image file type are TIFF, GIF, JPEG, WMF and PICT.

2.5 Edge detector

2.5.1 Introduction to Edge

In a picture, an edge is normally defined as an abrupt change in color intensity (*Internet Reference, 12/07/2004)*. Human uses eyes through stereoscopic vision and depth perception with incredible inference skills to find edges. This complicated method makes us easier to determine edges even if the object is overlapping between the each other.

Computer with vision system normally deals with grayscale camera. It must change the color intensity to find the edge of an image.

2.5.2 The edge detection

Edge detection techniques focus on identifying continuous adjacent pixels which differ greatly in color intensity because these are the one that mark the boundaries, between objects, or an object and the background, and it will form an edge. After the edge detection process is complete, edges can be easily identified.

Furthermore, edge and line detection filters subtract all parts of the image except edges or boundaries between two different regions. Edge detection is often used to recognized objects of interest in the image

There are various type of edge detector such as Canny Edge Detector, Laplacian Edge Detector and Sobel Edge Detector. The edge detector for this system is the Sobel edge detector.

2.5.2.1 Canny Edge Detector

What is Canny Edge Detector?

The Canny operator was designed to be an optimal edge detector (according to particular criteria - there are other detectors around that also claim to be optimal with respect to slightly different criteria). It takes as input a gray scale image, and produces as output an image showing the positions of tracked intensity discontinuities (*Internet Reference*, 27/09/2004).

How It Works?

The effect of the Canny operator is determined by three parameters:

i) The width of the Gaussian kernel used in the smoothing phase, and the upper and lower thresholds used by the tracker. Usually, the upper tracking threshold can be set quite high and the lower threshold quite low for good results. Setting the lower threshold too high will cause noisy edges to break up. Setting the upper threshold too low increases the number of spurious and undesirable edge fragments appearing in the output.

ii) Increasing the width of the Gaussian kernel reduces the detector's sensitivity to noise, at the expense of losing some of the finer detail in the image.

iii) The localization error in the detected edges also increases slightly as the Gaussian width is increased.

The problem with Canny operator is to do with Y-junctions *i.e.* places where three ridges meet in the gradient magnitude image. Such junctions can occur where an edge is partially occluded by another object. The tracker will treat two of the ridges as a single line segment, and the third one as a line that approaches, but doesn't quite connect to, that line segment. But this problem can be solved by including a model of such junctions in the ridge tracker. This will ensure that no spurious gaps are generated at these junctions.

2.5.2.2 Laplacian Edge Detector

What is Laplacian Edge Detector?

It is morphing algorithm which operates on features automatically extracted from target images. It is used to find the edges in the target images

The Laplacian method searches for zerocrossings in the second derivative of the image to find edges.

How It Works?

i) Selecting an image to perform edge detection.

ii) Blur the image – To perform a morph, it is only to select edges. But not every edge of the image, only the main features. Thus, the image is blurred prior to edge detection. This blurring is accomplished by convolving the image with a

gaussian (A gaussian is used because it is "smooth"; a general low pass filter has ripples, and ripples show up as edges).

iii) Perform Laplacian on this blurred image

An example of Laplacian operations on a signal:

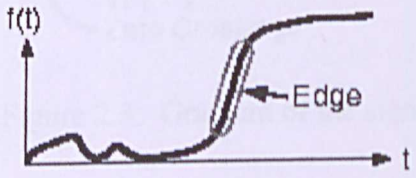


Figure 2.1: An example of one dimensional signal

The gradient of this signal (which, in one dimension, is just the first derivative with respect to t) is taken and get the following:

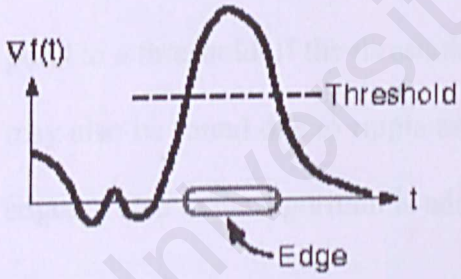


Figure 2.2: Gradient of the signal; first derivative with respect to t

Clearly, the gradient has a large peak centered around the edge. By comparing the gradient to a threshold, an edge can be detected whenever the threshold is exceeded (as shown above). In this case, the edge is found, but the edge has become "thick" due to the thresholding. However, since the edge occurs at the

peak, localization can be made by computing the laplacian (in one dimension, the second derivative with respect to t) and finding the zero crossings.

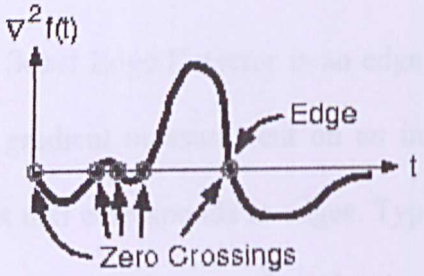


Figure 2.3: Gradient of the signal; second derivative with respect to t

The above figure shows the Laplacian of the one-dimensional signal. As expected, the edge corresponds to a zero crossing, but other zero crossings can be seen which correspond to small ripples in the original signal.

iv) Find the zero crossings of the Laplacian and compare the local variance at this point to a threshold. If the threshold is exceeded, declare an edge. But false edges may also be found due to ripple and texture in the image. To remove these false edges, a step to the algorithm is added.

v) Median Filter the image. Median filter is applied because it removes the spot noise while preserving the edges. This yields a very clean representation of the major edges of the original image.

2.5.2.3 Sobel Edge Detector : Chosen Edge Detector for the System

What is Sobel Edge Detector?

Sobel Edge Detector is an edge detector. The operator of Sobel performs a 2-D spatial gradient measurement on an image and so emphasizes regions of high spatial gradient that corresponds to edges. Typically it is used to find the approximate absolute gradient magnitude at each point in an input grayscale image (*Internet Reference*, 08/07/2004).

How it works?

The Sobel Edge Detector uses a simple *convolution kernel* to create a series of *gradient magnitudes*. Convolution K to pixel group p can be represented as:

$$N(x,y) = \sum_{k=-1}^{1} \sum_{j=-1}^{1} K(j,k)p(x-j,y-k)$$

It uses two convolution kernels, one to detect changes in vertical contrast (h_x) and another to detect horizontal contrast (h_y).

Below are the convolution kernels:

$$h_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad h_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

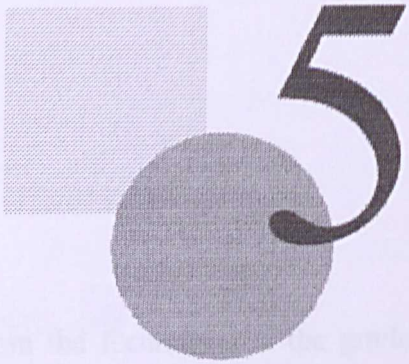


Figure 2.4: An example of a picture

Here is what the Sobel kernels do to the example picture (inversed to provide better clarity):

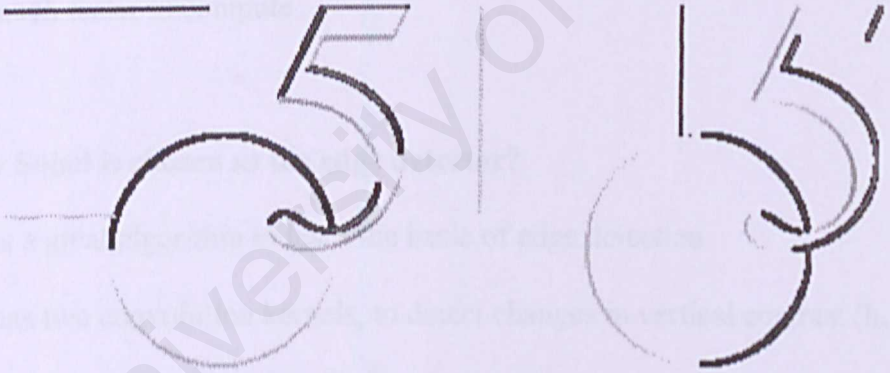


Figure 2.5: Image performed with Sobel

Data can now be represented as a vector (gradient vector) using Sobel. The two gradients computed using h_x and h_y can be regarded as the x and y components of the vector. Gradient magnitude and direction can be represented as below:

$$\mathbf{g} = \begin{bmatrix} g_x \\ g_y \end{bmatrix}$$

$$g = \sqrt{g_x^2 + g_y^2}$$

$$\theta = \tan^{-1}\left(\frac{g_y}{g_x}\right)$$

From the formula, **g** is the gradient vector, g is the gradient magnitude and □ is the gradient direction. All keen programmers will notice that it is probably more efficient to simple calculate the magnitude by adding the absolute squares - this is indeed what many implementation of the Sobel detector do. Typically, an approximate magnitude is computed using:

$$|g| = |g_x| + |g_y|$$

which is much faster to compute.

2.5.3 Why Sobel is chosen as the edge detector?

i) It is a great algorithm to learn the basic of edge detection

ii) It has two convolution kernels, to detect changes in vertical contrast (h_x) and to detect horizontal contrast (h_y) that suits with the following representation of image data, the Reduced Chain Code.

2.6 Shape representation for image data

In general, there are two categories of shape representation. There are the boundary-based and the region-based. But in this system, the shape representation that is used is the boundary-based representation. The basic chain code can be used for representing the boundary of any shape (Freeman & Davis, 1977). The chain code representation consist of tracing the boundary that compose the image, starting at initial pixel and then using numbers to describe the next point

The boundary can be traced in either a clockwise or a counter-clockwise manner and eight codes for every pixel are assigned according to the direction of the next pixel with respect to the current pixel. But basic chain code is not invariant to scale and rotation.

3	2	1
4	P	0
5	6	7

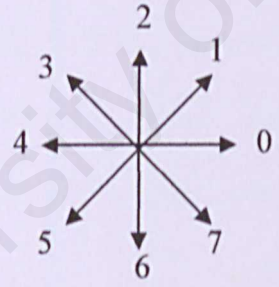


Figure 2.6: Chain Code -3x3 grid and direction codes

2.6.1 Reduced Chain Code

Since the ordinary chain code is not invariant to scale and rotation, modified chain codes is used for the purpose of shape representation. The modified chain code is the Reduced Chain Code.

The reduced chain code will be use in the system to represent image data for the boundary of image input to the system. It can differentiate the rotation and the scale of

any shape represented. There will be four numbers (0,1,2,3) used to code the boundary of an image according to the direction of the next pixel with respect to the current pixel.

The direction is described by each number in the reduced chain code below:

3	2	1
0	P	0
1	2	3

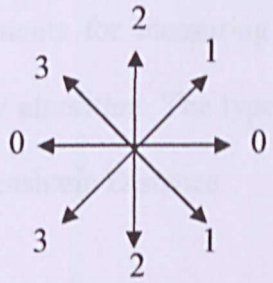


Figure 2.7: Reduced chain code -3x3 grid and direction codes

2.7 Matching using String Measure Technique

There are a few type of string measure technique such as the Nonlinear Elastic Matching, Extended Distance and Weighted Levenshtein Distance. All of this measurement is used as instruments for measuring shape similarity between imaged coded into strings of symbols by algorithm. The type of sting measure technique use in this system is the Weighted Levenshtein Distance.

2.7.1 Nonlinear Elastic Matching (NEM)

An interesting alternative measure is called nonlinear elastic matching. This measure does not obey the traditional metric properties but instead it is a relaxed set of metric properties. In practice, this provides the same advantages as any metric, and therefore can be used for clustering. However, the arbitrary number of points distributed on the edges, the need of determining certain starting matching points and the complexity of computing using dynamic programming makes this measure inappropriate for very large data sets.

Given two strings of images, $A= \{a_1,a_2,\ldots,a_m\}$ and $B=\{b_1,b_2,\ldots,b_n\}$, in order to define the NEM distance let one iteratively define quantities d(i,j), as

$$d(1,1) = h(1,1)$$

$$d(i,1) = d(i-1,1) + h(i,1) \quad i>1$$

$$d(1,j) = d(1,j-1) + h(1,j) \quad j>1$$

$$d(i,j) = \min \{d(i-1,j-1), d(i,j-1), d(i-1,j)\} + h(i,j) \quad i>1 \text{ and } j>1$$

and

$$h(i,j) = \begin{cases} 0, & \text{if } a_i = b_i \\ 1, & \text{otherwise} \end{cases}$$

The value of d(i,j) is a measure of the difference between the substrings $\{a_1,a_2,\ldots,a_m\}$ and $\{b_1,b_2,\ldots,b_n\}$. The NEM distance between strings A and B is defined as

$$D_{NEM}(A,B) = d(m,n).$$

2.7.2 Extended Distance

Extended Distance can be interpreted as a generalization of symbol rewriting rules into the following rules concerning groups of symbols:

$$a\,a\,b \rightarrow a\,b\,b\ (\alpha\, p)$$

$$a\,b\,b \rightarrow a\,a\,b\ (\alpha\ p)$$

$$a \rightarrow b\ (p)$$

$$a \rightarrow a\,a\ (\beta\ q)$$

$$\varepsilon \rightarrow a\ (q)$$

$$a\,a \rightarrow a\ (\gamma\, r)$$

$$a \rightarrow \varepsilon\ (r)$$

where a and b are generic symbols p, q, q are the substitution, insertion and deletion costs respectively; and α, β, γ are real numbers in [0,1], and ε is an empty symbol.

Letting

$$w_s = \begin{cases} 0, & \text{if } a = b_j \\ \alpha\, p, & \text{if } (a_{i-1} = a_i = b_{j-1} \quad \text{and } a_{i+1} = b_j = b_{j+1}) \\ & \text{or } (a_{i-1} = a_i = b_{j-1} = b_j \quad \text{and } a_i = a_{i+1} = b_{j+1}) \end{cases}$$

$$w_j = \begin{cases} p, & \text{otherwise} \\ \beta q, & \text{if } a_i = a_{j-1} = b_j \\ q, & \text{otherwise} \end{cases}$$

$$w = \begin{cases} \gamma r, & \text{if } a_{i-1} = a_i = b_j \\ r, & \text{otherwise} \end{cases}$$

the Extended Distance can be recursively defined as

$$D_{ED}(A,B) = l(m,n)$$

2.7.3 Weighted Levenshtein Distance

Weighted Levenshtein distance (WLD) is a one of the string measure distance technique. It measures the similarity between two strings. It computes the similarity between source and target features. According to their similarity value, matching process in the database is done and similar image is sort out. Then the similar images are displayed to the user.

WLD technique is based on the concept that string A can be rewritten into string B by means of one of three possible symbol operations, namely, symbol substitution, symbol insertion and symbol deletion. There is a generally more than one ways of rewriting string A into string B, each one characterized by the numbers s_k, i_k and d_k of the intervening substitutions, insertions and deletions: that are called N. If the costs relative to each operation are p, q and r, respectively, the WLD between A and B is defined as.

$$D_{WLD}(A,B) = \min_{k} \{s_k p + i_k q + d_k r\}$$

An iterative definition of WLD, useful for its practical implementation is

$$l(i,j) = \min \{ l(i-1, j-1) + w_s, l(i, j-1) + w_i, l(i-1) + w_d \}$$

for $1 \leq i \leq m$ and $1 \leq j \leq n$ with

$$.w_s = \begin{cases} 0, & \text{if } a_i = b_j \\ p, & \text{otherwise} \end{cases}$$

and boundary conditions

$$l(i,0) = ir \qquad i=0,1,2,\ldots,m$$

$$l(0,j) = jq \qquad j=0,1,2,\ldots,n.$$

Then it can be proven that

$$D_{WLD}(A,B)=l(m,n)$$

The image that will be retrieved is based on the comparison made between the Weighted Levenshtein Distance of the images. If the greater the Weighted Levenshtein Distance is, the more different the strings are and this means that the target image is not similar to the source image.

To retrieve the relevant or similar image to the one queried to the system, the source string must be identical or not so different with the target string. Then the output is images that have similar shape with the source image or shape chosen by the user.

2.8 Comparison with existing system

2.8.1 Image Edge Detection

This system is developed by Miss Azimah Razali, a student of Faculty of Science Computer and Information Technology, University of Malaya on the year 2002. The purpose of this system is to detect edge of digital images. It is software to detect edges and is a stand-alone system

Two main methods are used in this system are the Laplacian method and Gaussian. The Laplacian is a 2-D isotropic measure of the second spatial derivative of an image. Beside these two methods, Sobel, Robert and Prewitt Operators are also provided in this system with limited functionality.

The five main tasks in this system are input image by the user of the system, pre-procession in order to make sure the detection is accurate, detection process, preview displayed images and printing out the original image.

The expected users of this system are kindergarten teachers, pictures editor and student who studying image processing.

There is a similarity between the technique used in Image Edge Detection and our Image Retrieval System for Web that is to detect edges, Sobel Edge Detector is used.

The main difference between both of the system is that the Image Edge Detection is a stand-alone system while our system is a web-based system. Web-based system is more practical these days because many systems are develop for the use in the internet to make it easier for user to access and use it.

2.8.2 Blobworld Image Retrieval System (*Internet Reference*, 02/09/2004)

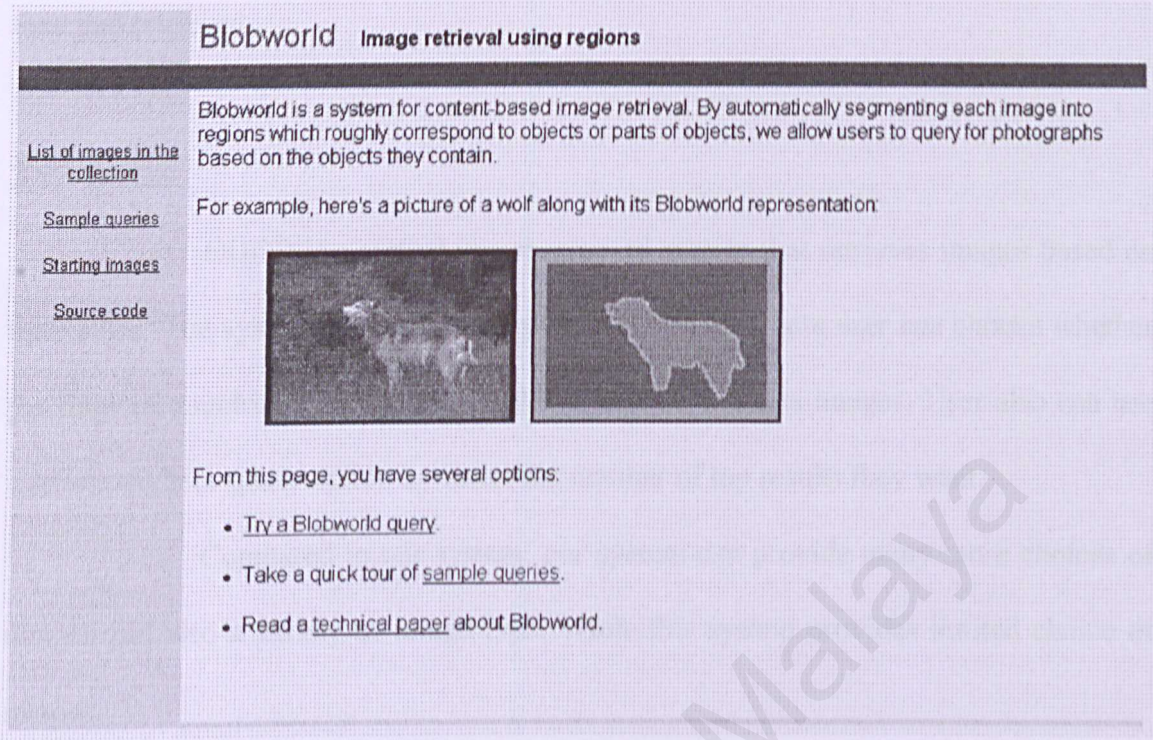


Figure 2.8: An example of the Blobworld Interface

Blobworld Image Retrieval System is online system for content-based image retrieval. It automatically segments each image into regions which roughly correspond to objects or parts of objects. This system allows users to query for photographs based on the objects they contain. The query is a combination of keyword and content-based searching. The user has options either to search images with content-based query or keyword query or both.

This system contains too many images in their image storage that might effect the accuracy of image retrieval where by too many images will be retrieved that is not related to their query. Another disadvantage of this system is that user cannot save or

download the images retrieved. Compared to our system, user can download images that they had retrieved.

2.8.3 OUDB

OUDB is a virtual image retrieval system that retrieves images based on their color. The system divides its query-menu into two where user can choose whether the flags of countries images or the college football helmets images. User also can use nearest neighbor query option by insert the number of top results they want.

Compared to our system, our system can provide many more choices of images and also a variety of image type. While this system provides limited choice of images.

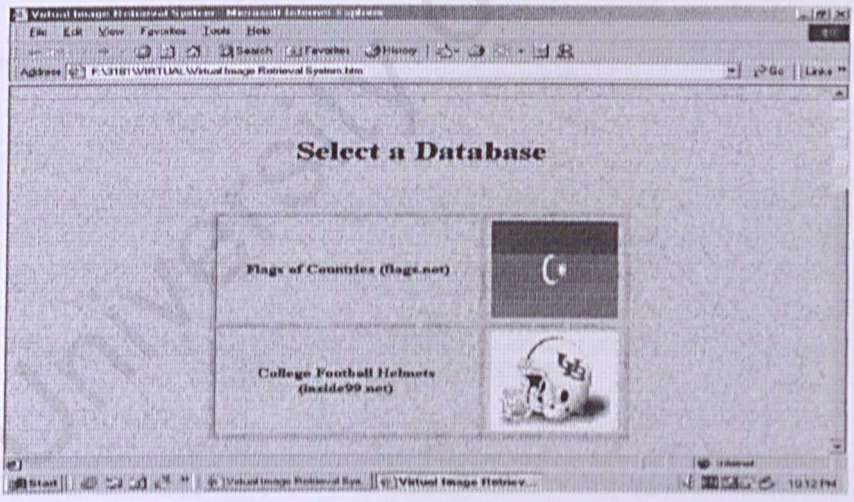


Figure 2.9: Main interface of the OUDB

2.8.4 CalPhotos

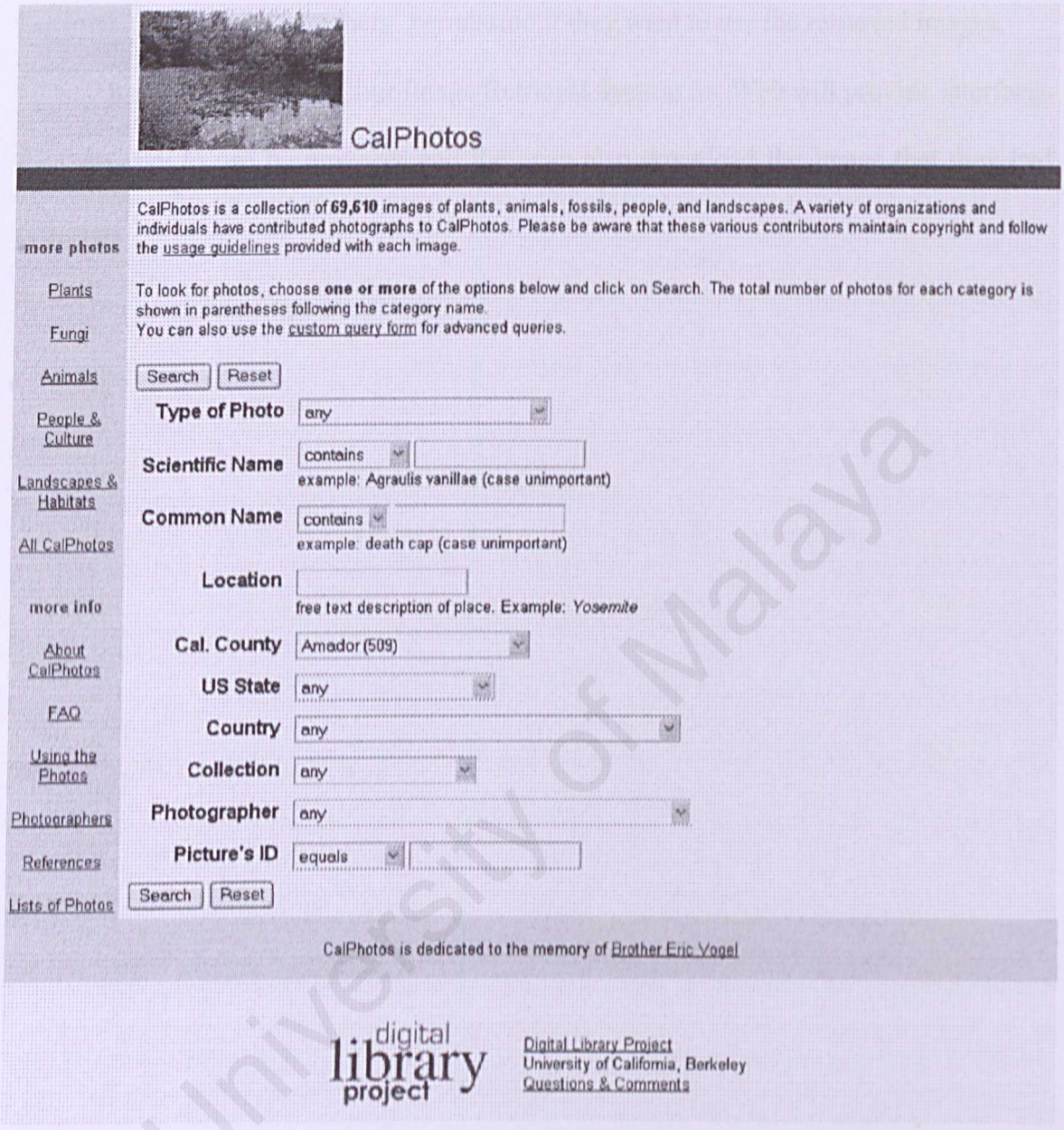


Figure 2.12: An example of the CalPhotos interface

Through this system, user can search for images. The user has to specify criteria of the images that they want to retrieve such as Type of Photo, Scientific Name cntain in the Photo and many more.

It provides a query interface that is hard to use by novice user. Besides that, user has to ask for the photographers' permission if they want to use the retrieved images.

But as a comparison, our Image Retrieval System for Web will provide interfaces that are easy to use by any user and they can also download the image that they had retrieved.

2.9 User Interface Design Technique

It is an important for the Image Retrieval System for Web to have an interface that is user friendly and easy to use. The issues to consider are to understand the users, their tasks and the technology to be use for the user interface.

The scope of user is for all types of people that have internet access as for the system is a web-based system. This means that the system is generally for the novice user.

The basic tasks involved in this system is querying, retrieving and downloading image. For the querying process, the users have to choose what type of query that they want to use. After the querying process, the next interface is the retrieval interface. The retrieved images that are similar to the query will be displayed to the user in the interface. After choosing an image from the images retrieved, the user can download the image from the download page.

The technology used to display the system must support graphical user interfaces. It can be either provided in the operating system or internet browser. Most browser run in most operating system. So for this system, platform independence must be considered. This means that instead designing interface for the chosen operating system, it is better to provide an interface that can operate in both dominating web browser that is the Microsoft Internet Explorer and Netscape Navigator.

2.10 Chapter Summary

This chapter consist the literature review for the Image Retrieval System for Web. The literature is about the techniques used to develop Module 1 and Module 2 in the system. The techniques involved in the review are Sobel Edge Detector, Reduce Chain Code, User Interface Design and Weighted Levenshtein Distance.

The explanation of tools and related technologies used in the system is also in this chapter. Among the explanation are the internet, web server, web browser, tools, programming language

There is also a comparison of existing systems that are Image Edge Detection, Blobworld, OUDB and CalPhotos. The comparison is only to discuss the advantages of the system to be developed compared to other systems.

CHAPTER 3:

METHODOLOGY

CHAPTER 3: METHODOLOGY

3.1 Introduction

Methodology means the science of method. It is a multiple step that will be a guide through the entire work in approaches that will effect on the final output of a product that is the Image Retrieval System.

In this system development, the methodology consist an approach to software development that is supported with a set of technique and notation, a life cycle model, and the support from unifying set of procedures and philosophy aid.

For this methodology, its aim is to make an efficient development cycle, high quality system, to get a fast turn-around at a low cost on system development.

3.2 System Development Methodology

3.2.1 System Development Life Cycle (SDLC)

The system to be developed must go through the System Development Life Cycle. SDLC is ongoing regarding the activities related to the system development. During the life cycle, there is a timeliness and deadlines until the system is installed and accepted by the user.

The five phases in the traditional SDLC are investigation, analysis, design, implementation, and maintenance and review are applied to the Image Retrieval System for Web.

- System investigation
  - Identify the potential problems and opportunities to define the system project and move on to the next phase.
- System analysis
  - When the problem is defined, the analysis is done to search for alternatives to solve it such as a study done on existing system. This will bring to a list of requirements for the Image Retrieval System for Web.
- System Design
  - In this phase, the technical design for the system is done to describe the new system.
- System implementation
  - From the assembling of the acquired or created system components detailed in the system design, the new system is put into operation.

❖ System maintenance and review

➢ When the new system is initialized, this phase makes sure that the system operates. Other than that, modifying will be done continuously to meet the changing business needs.

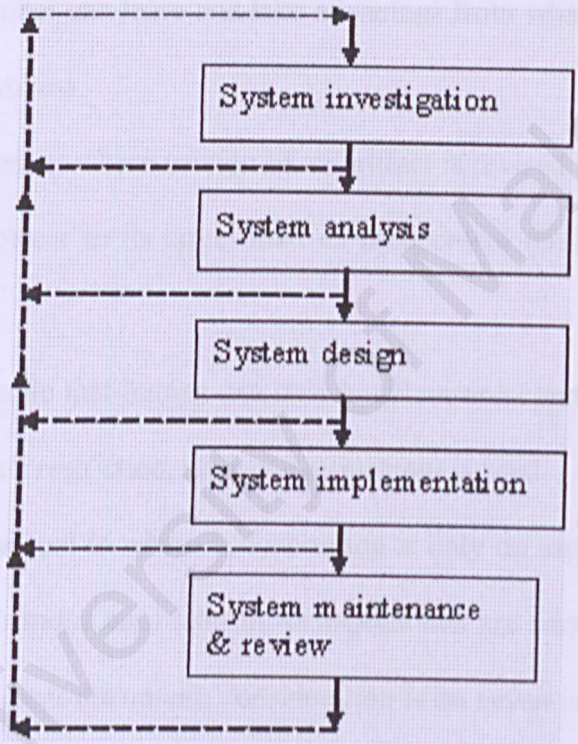


Figure 3.1: Traditional System Development Life Cycle

3.3 System Life-Cycle Model

In every system developed, there must be a model to represent the system life-cycle. For the Image Retrieval System for Web, the model of system life-cycle chosen is the Iteration-and-Incremental Model.

3.3.1 Iteration-And-Incremental Model

The basic idea of an iterative approach is to develop a system incrementally so that the system developer can learn and take advantage from what has been done earlier in the development process.

Iteration process is where version of an artifact is revised over and over again so that each version is closer to its target than its predecessor until reaching a satisfying version.

Miller's Law state that human can only handle seven chunks unit of information at a time. To handle this restriction, stepwise refinement is used.

Stepwise refinement is where concentration is only on aspects that are currently most important and postpone until later those aspects that are currently less critical. For example, for a requirement document, consideration is on seven important requirements until there is this next most-important requirement. It goes on and on which make it an incremental process.

In practice, an artifact is constructed piece by piece (incrementation) and each increment goes through multiple versions (iteration)

The five different workflows involved in the entire life cycle are:

- Requirements workflow
  - This is to determine what the user needs and their constraints.

- Analysis workflow
  - ♦ This workflow concentrates on analyzing and refining the requirements to know what the system is supposed to do.
- Design workflow
  - ♦ This is where detail description on how the system does all the functions of the system.
- Implementation workflow
  - ♦ Implementation of the target system is done in this workflow. The process is done where the system is partitioned into smaller subsystem to implement it in parallel.
- Test workflow
  - ♦ Test workflow is done throughout the whole system development.

From Figure 3.2, within each increment, the Waterfall Model repeats itself that make up the process of iteration. But the figure is not intended to be an accurate representation of precisely how the system product developed. Also from the figure, the number of increment does not have to be 4 increments and 3 iterations. It only shows the emphasis changes from iteration. This means that the number of increment and iteration for a system vary according to the use of it.

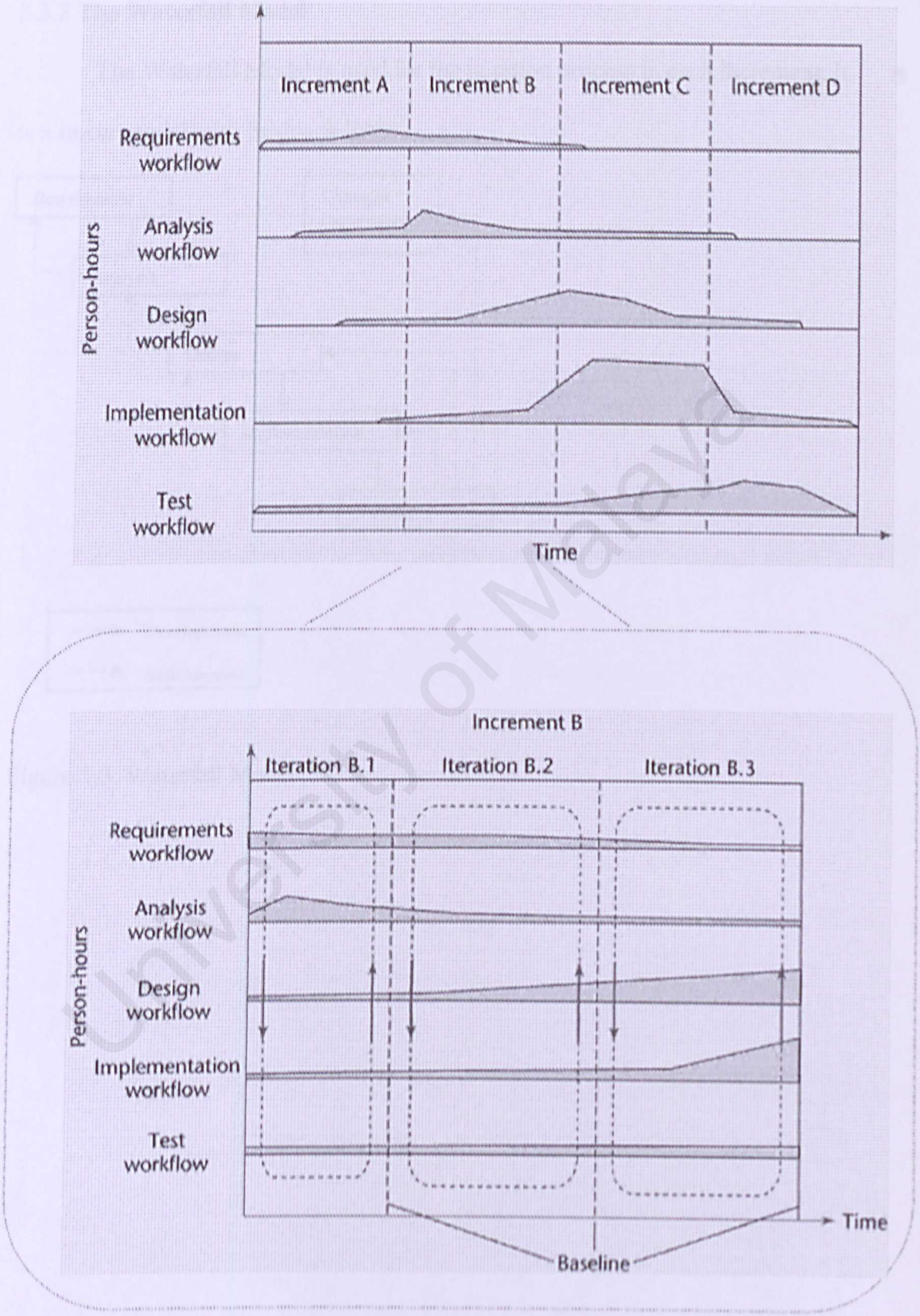


Figure 3.2: The Iteration-And-Incrementation Model

3.3.2 The Waterfall Model

The Waterfall Model is used for the iteration process in each increment. It is a linear model with feedback loops.

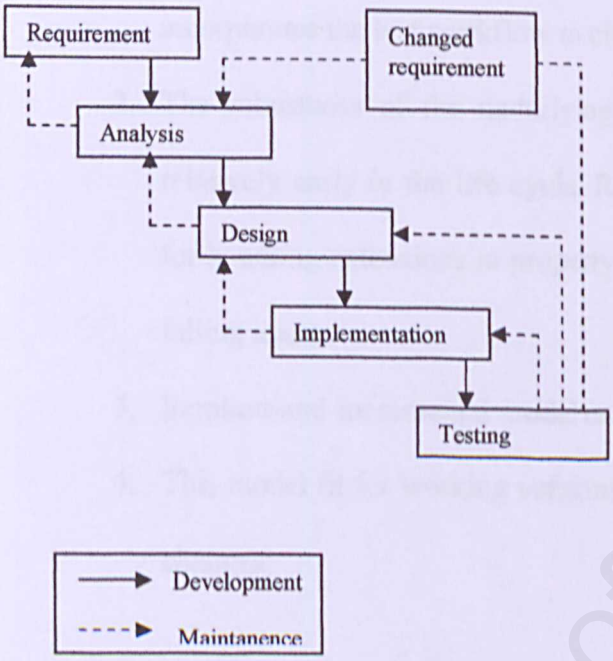


Figure 3.3: Waterfall Model

3.3.3 The advantages of Iteration and Incremental Model

1. Multiple opportunities are offered for checking that the system is correct. Faults can be detected earlier because every iteration incorporates the test workflow to check for error.
2. The robustness of the underlying architecture can be determined relatively early in the life cycle. Robustness is an important quality for handling extensions in property of a system and changes without falling apart.
3. Iteration-and-incremental model enables risk to be mitigated earlier.
4. This model fit for working versions of a system that keeps on having changes.

3.4 Chapter Summary

This chapter is about the methodology used in developing the Image Retrieval System for Web. In general, the System Development Life Cycle is explained because every system must go through this life-cycle. Then the model used for the system is the Iteration-and-Incremental Model. It is chosen as the model because it has many advantages. But the main advantage that it perform routine testing throughout the iteration and incremental development process so that correction on the error can be done earlier. This can reduce the cost of post delivery maintenance.

CHAPTER 4:

SYSTEM ANALYSIS

CHAPTER 4: SYSTEM ANALYSIS

System analysis is a technique to solve problem that decompose the system to be develop into component pieces. The purpose of this system analysis is to know how well those component parts work and interact to accomplish their purpose. For the Image Retrieval System for Web, the system analysis covers the functional requirement, non-functional requirement, software requirement and hardware requirement.

4.1 Requirement Analysis

Requirement analysis phase is also called definition phase or logical design phase. A requirement is a feature of the system or a description of what the system is capable of doing. And for that, the requirement analysis is done to fulfill the system purpose and define the requirement for a new system. The requirement analysis covers the area of functional and non-functional requirements of this project.

4.2 Functional Requirement

A functional requirement will describe an interaction between the system and its environments. It can predict how a system reacts in certain situation. Functional requirement are subsystems features and functions that must be included in an Information System and are frequently identified in terms of processors, outputs, inputs and stored data that are needed to satisfy the system needs and be accepted to the users. The functional requirements for the Image Retrieval System for Web are:

i) Access the System

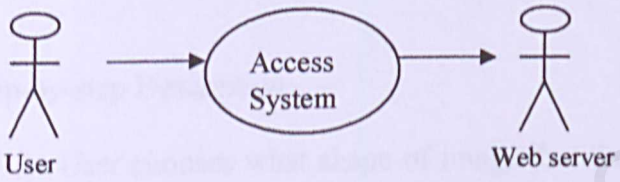


Figure 4.1: The Access the System Use Case

Brief Description:

The Image Retrieval System is an online system. It can be access by the user through the internet.

Step-by-step Description:

1. The user just has to type in the URL of the system
2. The web-server will define the website
3. The system will be displayed for the use of the user

ii) Request image

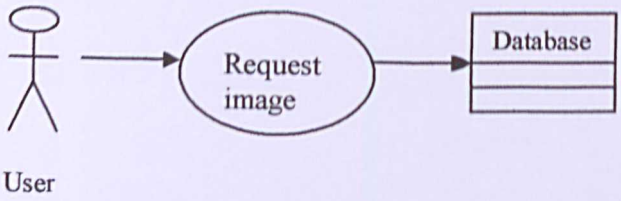


Figure 4.2: The Request Image Use Case

Brief Description:

The user can request the images using the system. From the main menu, the user can select what shape of image that they want to retrieve.

Step-by-step Description:

1. User chooses what shape of image that they want to request for from the menu button.
2. The system processes the query and sends it to the database.
3. The database returns images that are queried by the user.

iii) Input image

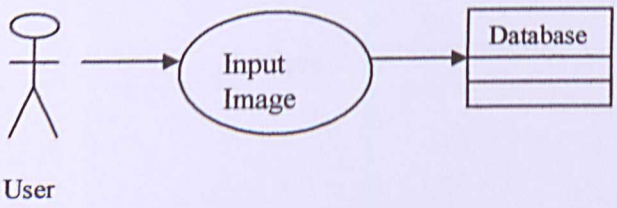


Figure 4.3: The Input Image Use Case

Brief Description:

The user will input the image to find images with similar shape.

Step-by-step Description:

1. From the main page of the system, the user browses for the image from any directory of the computer.
2. The selected the image is then analyzed in the system and transform to image data.
3. The data of the image is match with the data in the database.
4. The database returns images that have similar shape with the input image.

iv) Retrieve image

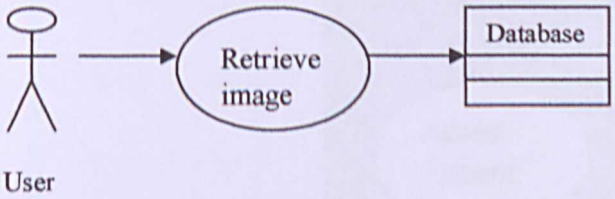


Figure 4.4: The Retrieve Image Use Case

Brief Description:

The user will retrieve images that are similar to their query.

Step-by-step Description:

1. The user queries the images either by using the menu button or by input image.
2. The system processes the query and sends it to the database.
3. Database then returns the similar shape images to the user through the system.

v) Download image

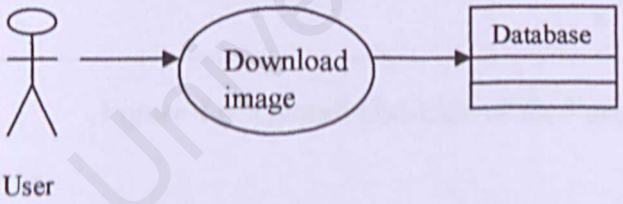


Figure 4.5: The Download Image Use Case

Brief Description:

After the user retrieved all the requested images, the user can download the image.

Step-by-step Description:

1. The user click on the download button provided in the interface to download the image retrieved.

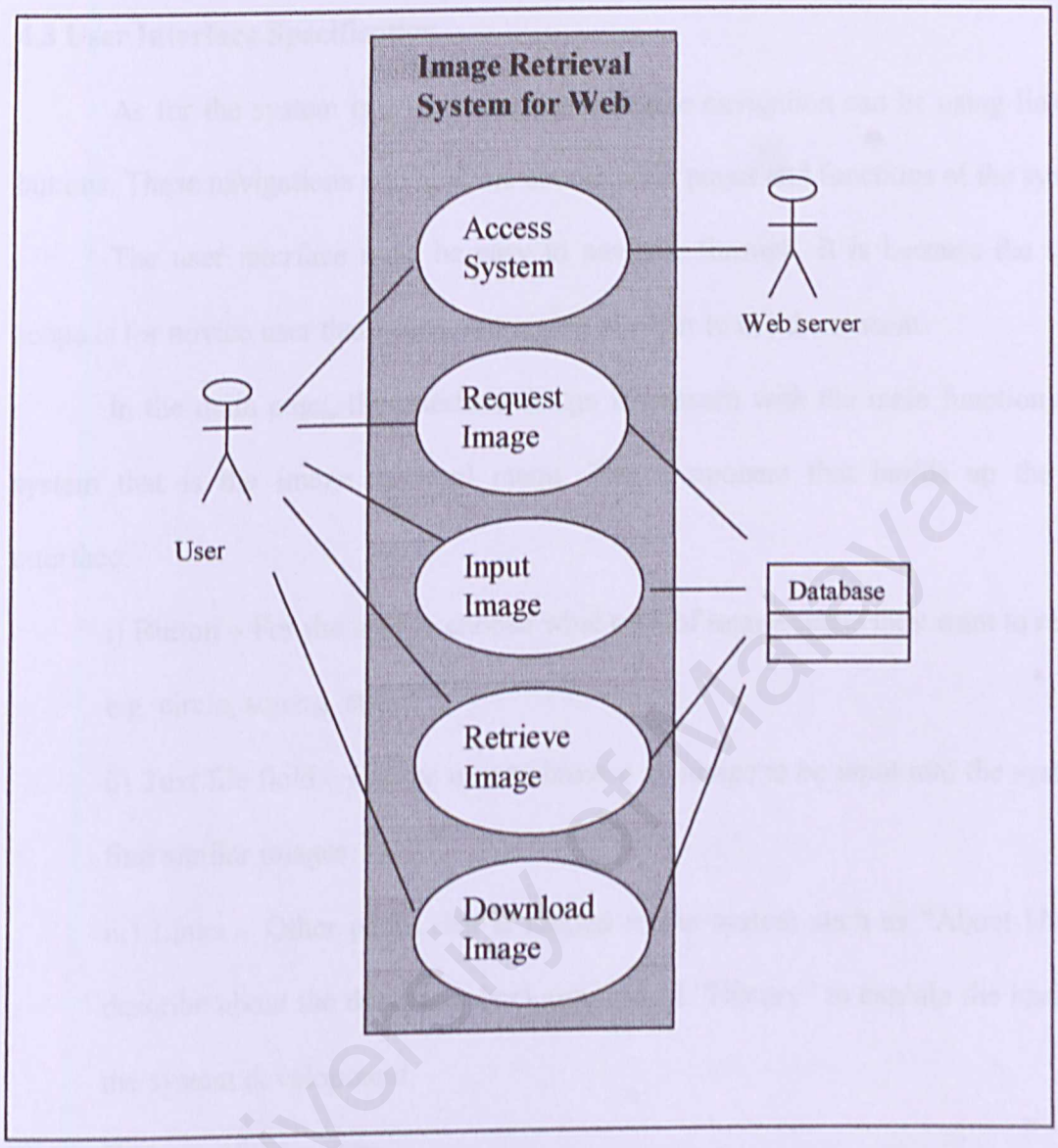


Figure 4.6: Overall Use-case of the Functional Requirements

4.3 User Interface Specification

As for the system is a web-based system, the navigation can be using links and buttons. These navigations will lead the user to other pages and functions of the system.

The user interface must be easy to navigate through. It is because the system scope is for novice user that needs no training in order to use the system.

In the main page, the interface design is concern with the main function of the system that is the image retrieval menu. The component that builds up the main interface:

i) Button – For the user to choose what type of image shape they want to retrieve e.g. circle, square, etc.

ii) Text file field – For the user to browse an image to be input into the system to find similar images.

iii) Links – Other pages that is related to the system such as "About Us" that describe about the developer background and "History" to explain the history of the system development.

Screen Formats of the system

Information

- Statement to be delivered to users
- E.g.: In the system, information provided is in the 'About Us' links whereby users can access overall description and information about the developer

Navigation

- Allow the user to be able to access and navigate through the system
- It also makes the expert user felt that they are controlling the system
- E.g.: Links to direct the user from one page to another page.

Action

- Involves action or interactivity in the system

4.4 Non-functional requirement

The non-functional requirement is to describe the constraints on the system. It is important so that the system is easy to use, operate as intended, provides output that is reliable to the user and can be maintained from time to time so that it copes with the evolving environment of web-based system.

i) Downloading time

The system involves a lot of images. So the downloading time for this system is a bit low especially when displaying images. But this problem can be solved by using suitable type of image type such as gif. This image file type can be used because it has an interlacing property – image blur at the time of downloading.

ii) Reliability

The system should provide a system that is reliable for the user. It should be able to retrieve similar images that are queried and accurate to return minimal false result.

iii) Ease of use

The user of the system could be anyone who can access the internet and into the system. So the system must be easy to use for novice user and need no training. This can be achieved with the design of user friendly interface.

iii) Maintainability

The system will be maintained from time to time by the system developer or the administrator. Maintenance is important when error encountered and needed to be fixed.

iv) Accuracy

The system needed accuracy on the image retrieved. Accuracy is when the number of image with similar shape is high in the retrieved images. The system can be a reliable system if the retrieval is accurate.

4.5 Software Requirement

Table 4.1: Software Requirement

Software Type	Software Name
Operating System	Windows XP
Application	Macromedia Dreamweaver MX
	Microsoft Front Page
	Adobe Photoshop
	Internet Explorer
Database	SQL Server
Programming language	ASP.NET

4.6 Hardware Requirement

Table 4.2: Hardware Requirement

Hardware Type	Specification
Processor	Pentium® 4 CPU 2.5 GHz
Memory	224 MB of RAM
Hard Disk	40 GB
Input Device	Keyboard, Mouse, Scanner, Digital Camera

4.8 Chapter Summary

This chapter is to do analysis for the system to know what is required. The system covers functional, non-functional, system, software and hardware requirement. The system analysis is important because it enables the developer to determine each part or component works and can be composed to build a system.

CHAPTER 5:

SYSTEM DESIGN

CHAPTER 5: SYSTEM DESIGN

In this chapter, system design is the specification of the detailed Image Retrieval System computer-based solution. The system design would be explained in drawings. This type of design is called the Model-Driven Design. It emphasizes drawing system models to document technical and implementation aspects of a system. The model-driven design approaches used in this system are Modern Structured Design, Information Engineering, Prototyping and Object-Oriented Design.

Modern Structured Design

- It is a system design technique that decomposes the system's processes into manageable components.

Information Engineering

- It is a model-driven and data-centered but process-sensitive technique. In this system, this technique is use to plan, analyze and design and information system.

Prototyping

- This is a modern-based approach to design. It is an iterative process involving the relationship between the designers and the users.

Object-oriented Design

- This type of design used technique to refine the object requirements definitions identified earlier during the analysis and to design-specific objects.

5.1 System network design of Image Retrieval System

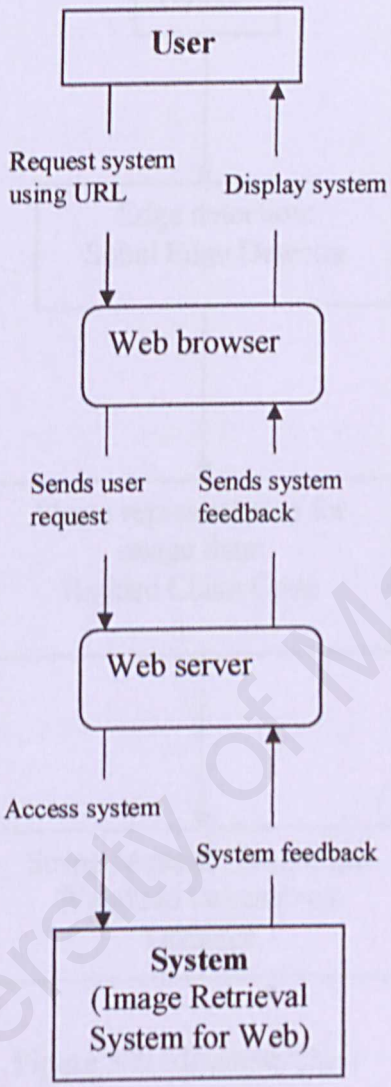


Figure 5.1: System network design

This system network design depicts the overall process of accessing the Image Retrieval System for Web. The user will type in the systems' URL and the web browser will send the user request to the web server. Then the web server will access the system. The feedback from the system will enable the user to use the system.

5.2 Structure Chart of Image Retrieval System

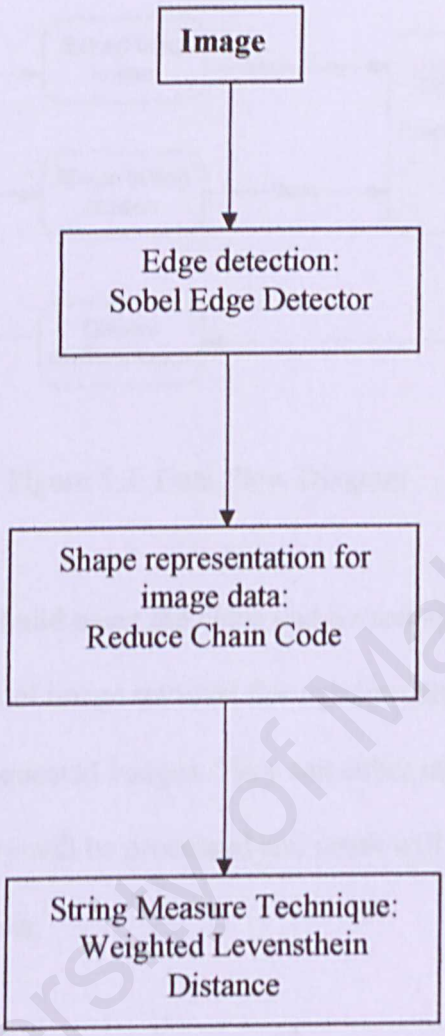


Figure 5.2: Structure Chart

Structure chart depict the process of the image analyzing, coding and matching. Sobel is used for the analyzing and detecting the edge, the image data will be represented using Reduced Chain Code and the Weighted Levensthein distance is used as a string measure technique.

5.3 Data Flow Diagram of Image Retrieval System

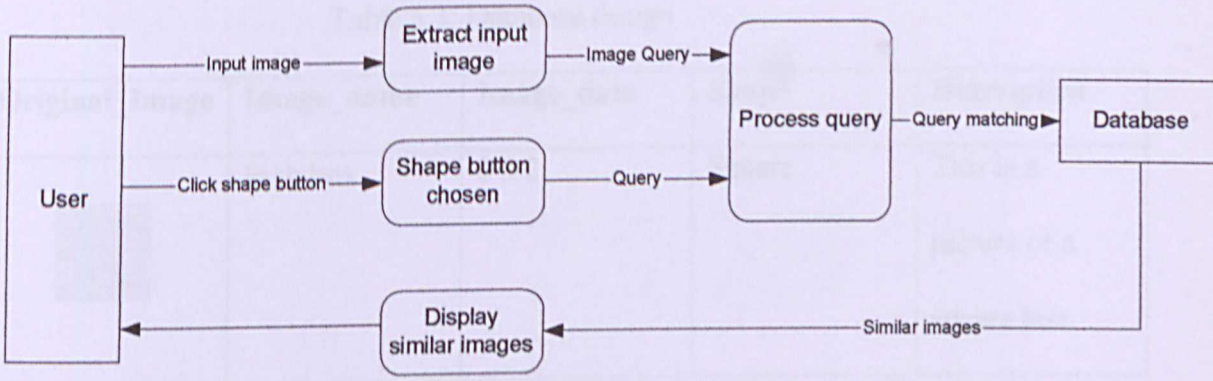


Figure 5.3: Data Flow Diagram

This Data Flow diagram is build using the Gane and Sarson approach. The overall diagram depicts the process of image retrieval through the database. There are two ways for the user query for the requested images. They can either input an image or click on shape button. Then the query will be processed and result will be similar images that would be displayed to the user.

5.4 Database Design

Table 5.1: Database design

Original_Image	Image_name	Image_data	Shape	Description
	Pink box	0 3 2	Square	This is a picture of a square box

This is the example design of the database. It would only contain a table to store the images and the details. The attributes are: Original_Image, Image_name, Image_data, Shape and Description.

5.5 User Interface Design for Image Retrieval System

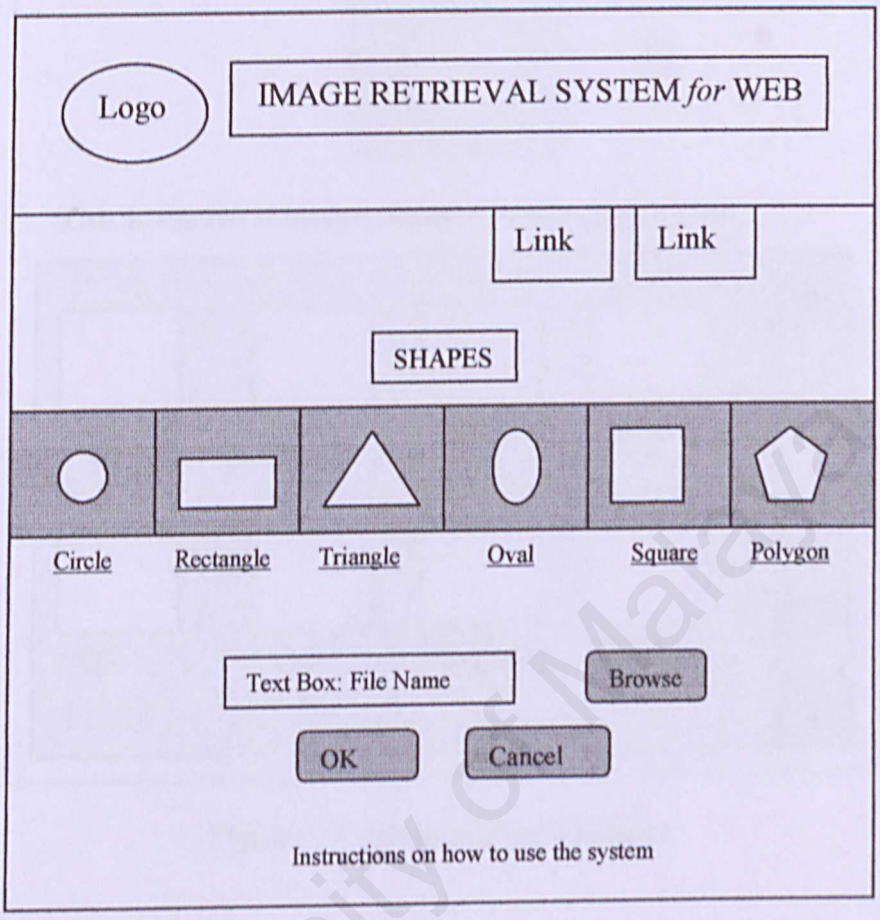


Figure 5.4: Main Page Interface

Figure above is the user interface design of the Image Retrieval System for Web. A brief description about this design is that this would be the baseline our main page of the system. The functions in this system would be displayed in the main page such as the shape buttons and the file text box to input the image by the user.

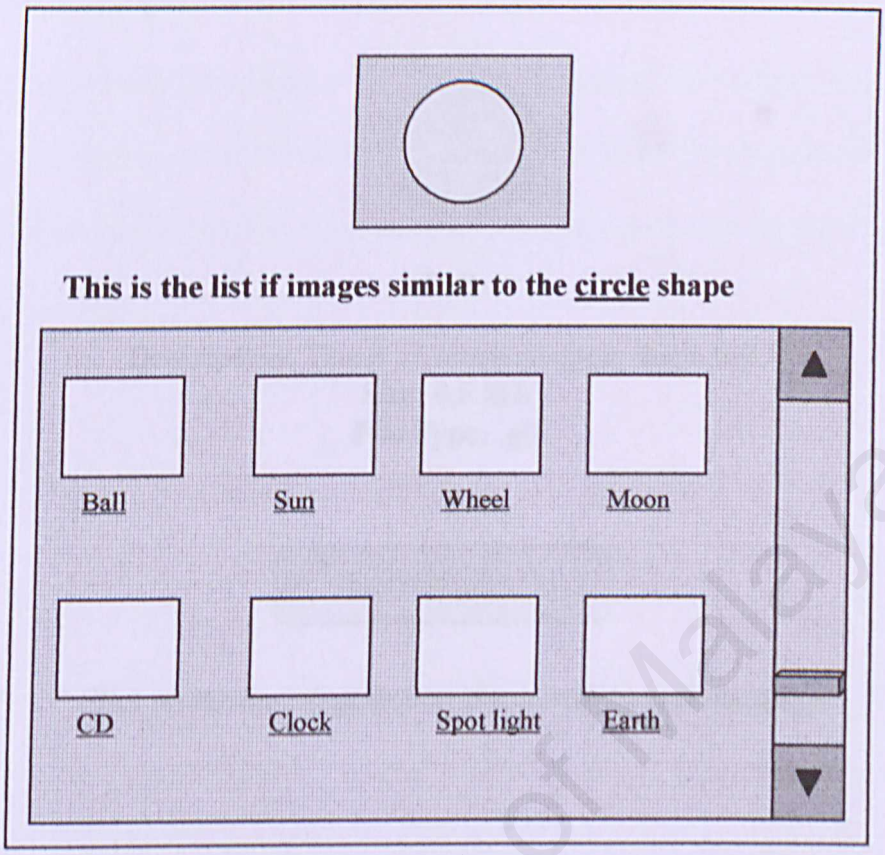
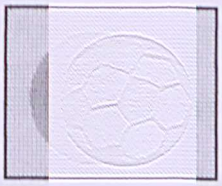


Figure 5.5: Retrieval Page Interface



Ball

Description: This is an image of circle shape ball.
Size: 0.5 MB
File Type: .gif

DOWNLOAD

** Click on the download button if you wish to save the image*

Figure 5.6: Download Page Interface

5.6 Chapter Summary

The system design consists of figures and description of the figure that are related to the Image Retrieval System for Web. The design involves system network design, structure chart, data flow diagram, database design and user interface design.

CHAPTER 6:

SYSTEM IMPLEMENTATION

CHAPTER 6: System implementation

6.1 Introduction

Implementation is the process of translating the detailed design into code. In chapter 5, the design had been made, and in this chapter will explain more about the process of assuring that the system can be translated into code from the design and executed as a program. System implementation implements the various components of the system based on the collected requirements, where the design is translated into a machine-readable form.

6.2 Development Environment

Development environment explains about the hardware, software and platform involved in the system implementation.

6.2.1 Hardware used in the system implementation

Table 6.1: List of hardware

No.	Name	Type
1	Processor	Intel Pentium 4
2	RAM	256 MB
3	Storage	40 GB
4.	Other standard desktop PC component.	USB Port, CD-Writer

6.2.2 Software used in the system implementation

Table 6.2: A list of software used

No.	Name	Type
1	Operating System	Microsoft Windows XP
2	Web server	Microsoft Internet Information Service V5.1
3	Database Management System	SQL Server 2000
4	Web Development Tool	Microsoft Visual Studio.NET 2003
	Coding Language • User Interface • Server Side Scripting • Client Side Scripting	 ASP.NET VB.NET VB.NET
5	Graphic Creation	Adobe Photoshop 6.0, Microsoft Paint
6	Web Browser	Microsoft Internet Explorer 6.0
7	Documentation	Microsoft Office

6.3 Platform Environment

To start the system development the platform environment must be prepared beside the hardware and software used.

The platform development will include setting up the Microsoft SQL Server 2000, configure IIS server, and install Microsoft Visual Studio.NET 2003.

6.3.1 Setting Microsoft SQL Server

The SQL Server is installed as client manager. To use the Enterprise Manager with the LOCALHOST as server, the SQL Server Setup form the MSDE must be installed.

In SQL server, Enterprise manager tool is used to create system database and all of the associated objects such as tables, views, diagrams and other.

For the system development, the Enterprise manager tool is used to create the Image Retrieval System database named IRS. Then tables created in the database. One example of table created is ImageGallery table.

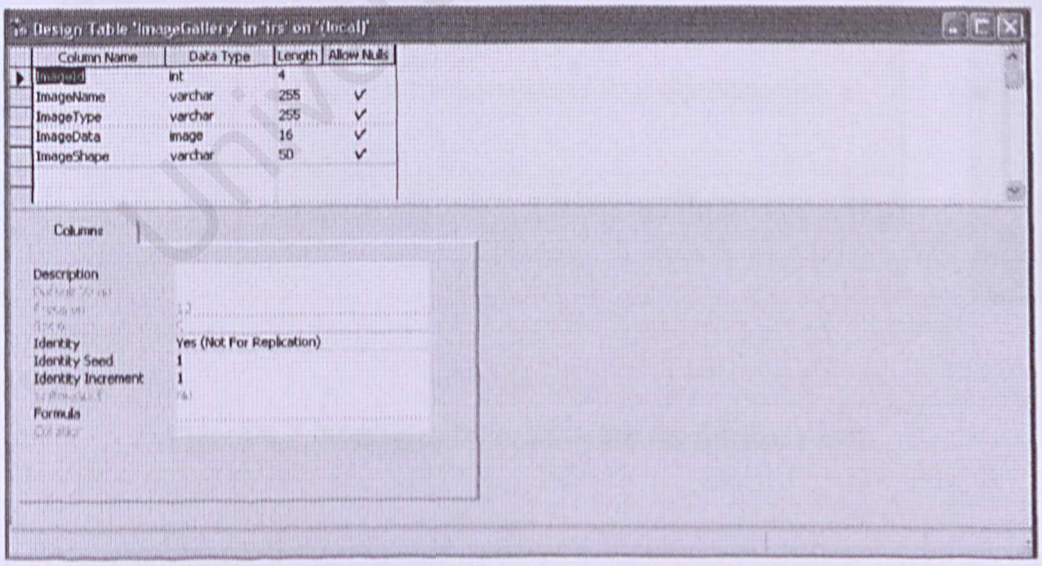


Figure 6.1: The design view of the ImageGallery table in the IRS database

To enable the administrator to gain access to the IRS database, the user of the database must be set. After the new database user had been created, the permission to SELECT, INSERT, UPDATE and DELETE is selected to enable the administrator to upload, view and delete images through the user interface of the Image Retrieval System for Web.

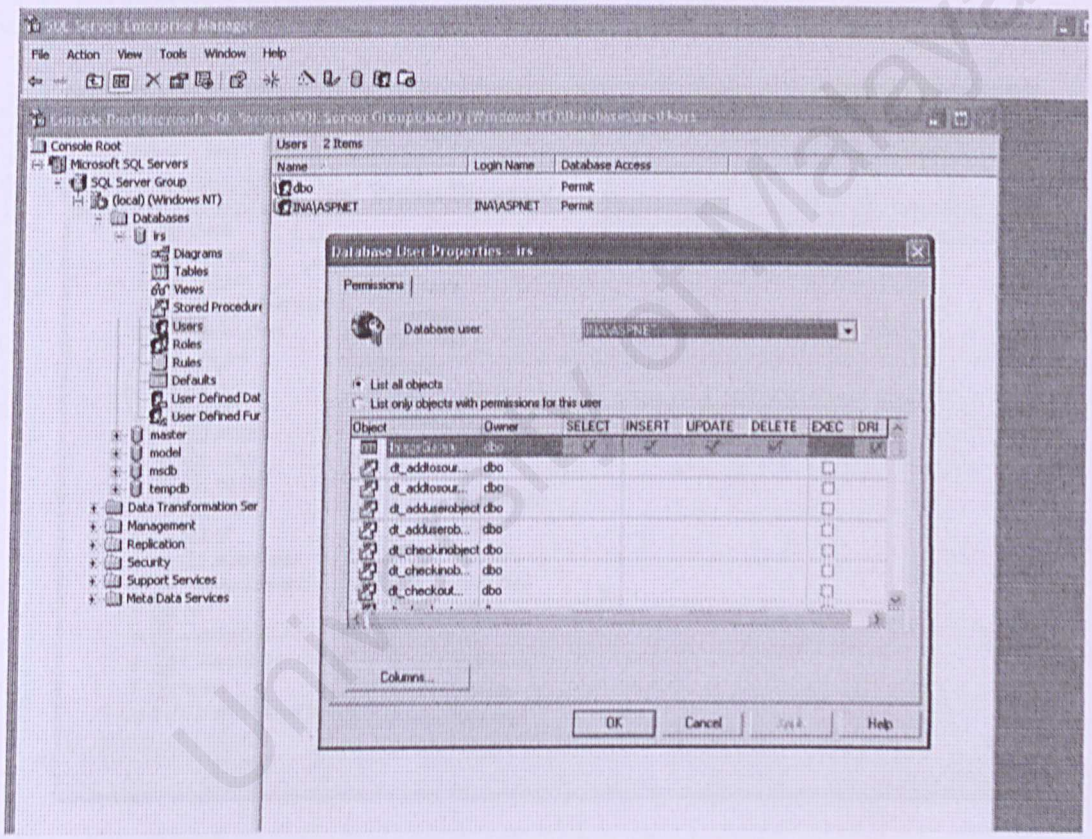


Figure 6.2: Managing Permission for the database user.

6.3.2 Configure IIS v5.1

The IIS is to be installed. After the IIS is installed successfully, new virtual directory is created in the Default Web Site. For the system development, virtual directory for the Image Retrieval System for Web named 'irs' is created.

The virtual directory will allows users access the application through HTTP protocol using web browser. The root directory for the IRS application is http://localhost/irs/

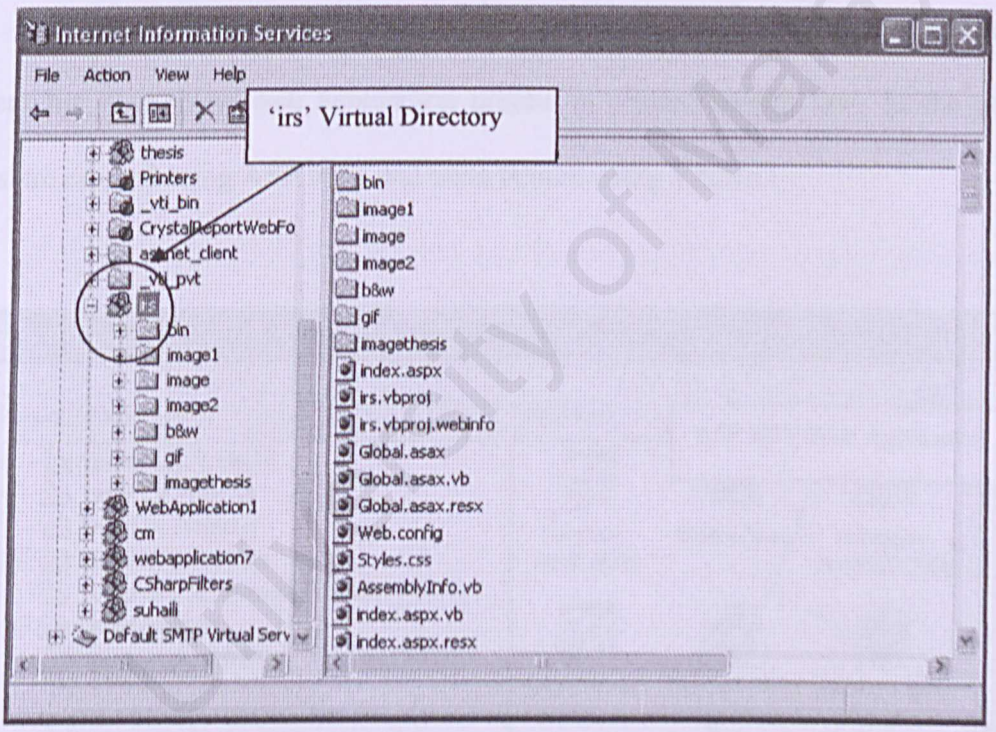


Figure 6.3: Virtual Directory for the Image Retrieval System for web; irs

6.3.3 Install Microsoft Visual Studio.NET 2003 and set the IRS web application project.

The.NET framework is the latest technology and it has also become the established way to build dynamic web pages and web-based application on a windows Server platform. The Microsoft Visual Studio.Net 2003 is the most apparent development tool for working with ASP.Net. Thus, Microsoft Visual Studio.NET 2003 is core tool of the project implementation that provides a whole range of editing, debugging and code building tools.

Microsoft Visual Studio.NET 2003 is used to develop the Image Retrieval System for Web. New web application project is created named 'irs'. In the project, pages are created using ASP.NET and code behind using VB.NET.

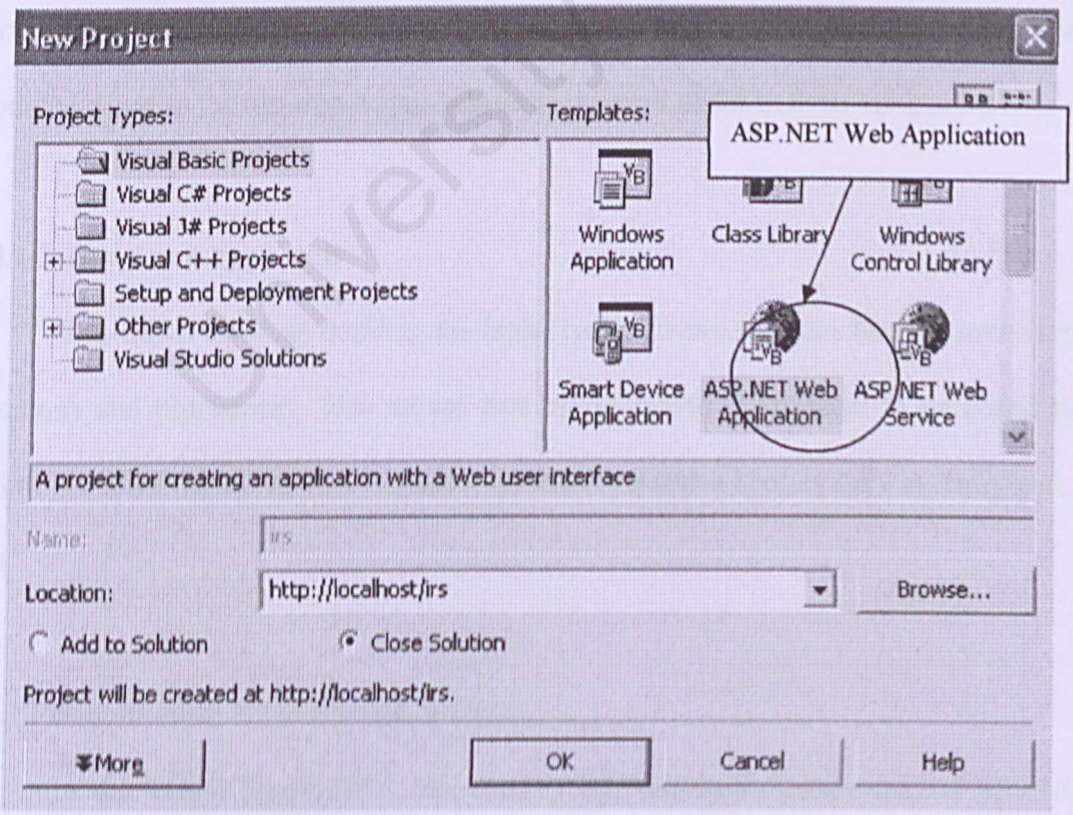


Figure 6.4: How to create web application in Microsoft Visual Studio.NET 2003.

6.4 Coding Environment

The coding environment is developed using the Microsoft Visual Studio.NET software. It involves types of coding approach used, programming language used and coding facilities.

6.4.1 Coding Approach

There are two types of coding approach: top-down and bottom-up. The Image Retrieval System for Web was developed modularly using both coding approach.

6.4.1.1 Bottom-up approach

The bottom-up coding is based on coding the lower-level modules initially and leaving the high level modules merely as skeletons that are used to call the lower modules whereas the top-down approach is the reverse of the bottom-up approach.

6.4.1.2 Top-down approach

Top-down involves building the high level software modules that are refined into functions and procedures. This means that the higher-level modules to be coded first before the lower modules. The codes in the lower modules contain only an entry and an exit.

6.4.2 Programming language use to code

In the Visual Studio.NET, two main programming language used is ASP.NET and VB.NET.

6.4.2.1 ASP.NET

The ASP.NET is a programming language used to develop the interface of the Image Retrieval System. It is used in the system development because it can be used to author applications in any .NET compatible language, including VB.NET. This programming language is embedded in the HTML code.

Each Web Form starts and ends with ASP.NET tag. The following figure depicts the use of ASP.NET to code the interface.

```
<asp:datalist id="DataList1" style="Z-INDEX: 101; LEFT: 160px; POSITION: absolute; TOP: 256px"
    runat="server" Font-Names="Verdana" RepeatColumns="3" repeatdirection="Horizontal" Height="96"
    Width="384">
    <ItemTemplate>
        <TABLE border="1" bgcolor="#ffffff">
            <TR>
                <TD width="150" height="100"><a href='<%#"Imager2.aspx?Source=DB&id="
                & DataBinder.Eval(Container.DataItem,"imageID")%>' target=_blank>
                <IMG  src='<%#"Imager.aspx?Source=DB&id=" & DataBinder.Eval(Container.DataItem,"imageID")%>
                </TD>
                <TD width="150" height="100"><a href='<%#"Imager2.aspx?Source=DB&id="
                &Dat                    er.DataItem,"imageID")%>' target=_blank>
                <%#c                    mageName")%></a></TD>
            </TR>
        </TABLE>
    </ItemTemplate>
</asp:datalist><asp:label id="Label1" style="Z-INDEX: 104; LEFT: 184px; POSITION: absolute; TOP: 16px"
runat="server"
    Font-Names="Showcard Gothic" Height="24px" Width="392px" ForeColor="Black" Font-Size="XX-Large">
    Upload Image</asp:label><INPUT id="UploadFile" style="Z-INDEX: 102; LEFT: 288px; WIDTH: 256px;
    POSITION: absolute; TOP: 112px; HEIGHT: 24px"
    type="file" size="23" name="UploadFile" runat="server">
```

ASP.NET Tag

6.4.2.2 VB.NET

VB.NET is used to support the functions written for the pages in Image Retrieval System. It is written as code-behind of every page. The following example explains the use of VB.NET to support the function of a page.

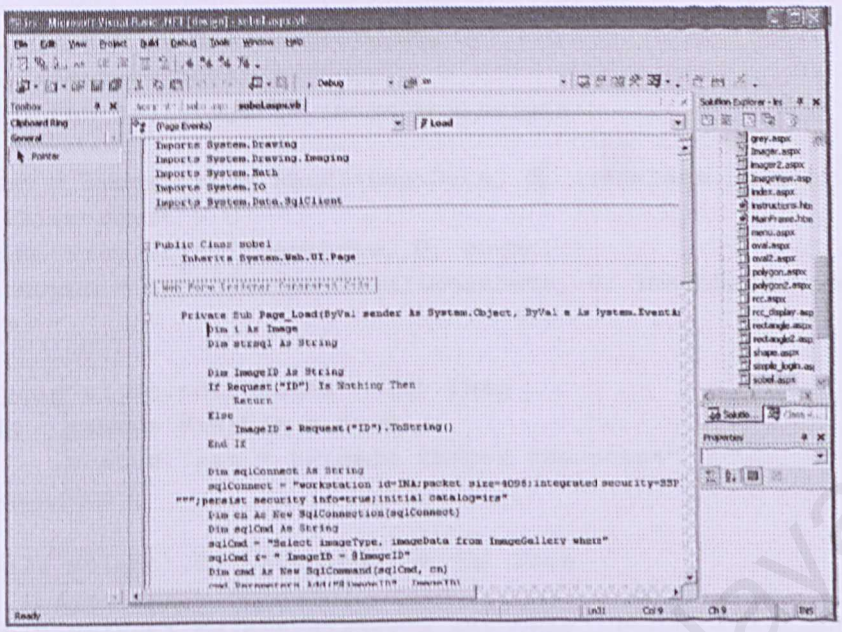


Figure 6.5: Example of Code-behind for Image Retrieval System in Microsoft Visual Studio.NET 2003

6.4.3 Coding facilities

Microsoft Visual Studio.NET provides some coding facilities to assist programmer in coding the system.

6.4.3.1 Formatting and indenting codes

Microsoft Visual Studio.Net 2003 provides user a good formatting and indenting facility where user is associated to format and indent the codes automatically. It is important to make it easier for the user to write the codes.

```
Dim mStream As New MemoryStream
Dim imageData(1024) As Byte
Dim count As Integer = imageData.Length

count = imageStream.Read(imageData, 0, imageData.Length)
Do While count > 0
    mStream.Write(imageData, 0, count)
    count = imageStream.Read(imageData, 0, imageData.Length)
Loop

If select1.SelectedIndex = (0) Then
    status.Visible = True
    status.Text = "Please choose the shape"
    Return

ElseIf select1.SelectedIndex = (1) Then
    imageshape = "circle"
ElseIf select1.SelectedIndex = (2) Then
    imageshape = "square"
ElseIf select1.SelectedIndex = (3) Then
    imageshape = "oval"
ElseIf select1.SelectedIndex = (4) Then
    imageshape = "rectangle"
ElseIf select1.SelectedIndex = (5) Then
    imageshape = "triangle"
ElseIf select1.SelectedIndex = (6) Then
    imageshape = "polygon"
End If
```

Automatically indented

Figure 6.6: Example of code indented automatically

6.4.3.2 Commenting Codes

Comment code is another important element in writing program. This helps the author to understand what and why the coding was written. This also makes it easier for other people especially collaborative programmers to understand the code. Comment code is not run in the program.

Comments can be included anywhere in the codes. For Microsoft Visual Studio.Net 2003 the single quotation mark (') is used as the prefix of a comment.

The following figure depicts the comment used in the code behind of Microsoft Visual Studio.NET 2003:

```
Function graypixel(ByVal b As Bitmap, ByVal x As Integer, ByVal y As Integer)
    Dim addRGB As Integer
    Dim tmpColor
    addRGB = 0

    ' for each pixel
    For y = 3 To b.Height - 1
        For x = 3 To b.Width - 1

            ' average out the rgb
            tmpColor = b.GetPixel(x, y)
            addRGB = tmpColor.r
            addRGB += tmpColor.g
            addRGB += tmpColor.b
            addRGB = addRGB / 3


            ' then Set the values
            b.SetPixel(x, y, Color.FromArgb(addRGB, addRGB, addRGB))

        Next
    Next

    Return b


    ' tidy up
    b.Dispose()


End Function
```

Comment Codes using single quote (')

Figure 6.7: Example of comment code

6.4.2.3 Error detection

Microsoft Visual Studio.NET also provides the capabilities of codes error detection. A line will be displayed under any code that contains error and a brief description of the error (on mouse over the error). The following code is an example of error code detected.

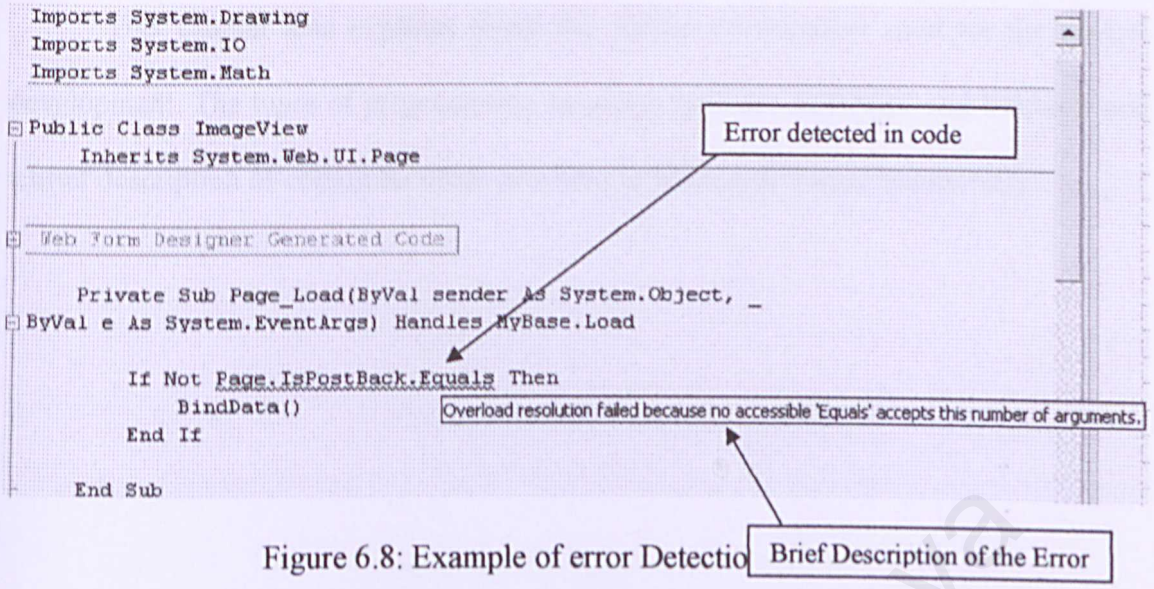


Figure 6.8: Example of error Detectio

6.5 Chapter Summary

This chapter explains about the various component involve in the system implementation. The main parts are hardware used, software used and the platform environment.

Hardware used to implement the system is the list of specification for the computer and other related component. The specification is high in order to be compatible with the software used to develop the system.

Software for the system development environment includes software for operating system, web server, database management system, web development tools, graphic creation, web browser and documentation software.

The platform environment is description about the setting up and configuration of the SQL Server, Internet Information Service (IIS) and Microsoft Visual Studio.NET.

This chapter also explains about the coding environment used for the system development. The types of programming language used are ASP.NET and VB.NET and a brief description of coding facilities provided in Microsoft Visual Studio.NET

CHAPTER 7:

SYSTEM TESTING

CHAPTER 7: System Testing

7.1 Introduction

The Image Retrieval System for Web is a system that is intended for public consumption should receive some level of testing. A number of different types of testing have to be performed. Without testing there is no assurance that the system developed will behave as expected.

System testing is carried out once the program components are coded. In this phase, tests are conducted to check the system modules for their consistency and error, plus ensuring that the system fulfill all the requirements and objectives.

7.2 Types of faults

The codes that had been implemented will be analyzed to detect the faults. These faults will have to be corrected immediately. The types of faults that usually exist in when developing a system are algorithmic faults and syntax faults.

7.2.1 Algorithmic faults

This type of faults occurs when the output is not as intended by the developer. The algorithmic faults happen because the program is not correct. To detect this fault, the developer has to simply read through the program code to know where did this fault occur or try submitting data into the system that would invoke the fault.

7.2.2 Syntax Faults

Syntax faults occur when the syntax of the programming language used is not correct. The result of this fault can sometimes bring effect the overall function written. With that, the developer must be careful to use only correct syntax.

7.3 Developing a test plan

The first step in testing is developing a test plan based on the requirement of the system. This plan usually done in a formal document that ensures the system meets the following standards:

i. Tested thoroughly – Untested code adds an unknown element to the product and increases the risk of product failure.

ii. Meets with the requirement made earlier – the system that meets the customer needs must provide the features and behavior described in the specification.

iii. Does not contain any faults – features must work within established quality standards, and those standards should be clearly stated in the test plan.

The test plan must include all of the following steps:

i. Establishing the test objectives

ii. Designing the test cases

iii. Writing the test case

iv. Testing the test case

v. Executing the tests

vi. Evaluating test result

7.4 Types of Tests

There are different types of tests that will be performed to ensure system that is implemented meets the overall requirement and does not contain ant defects. Types of test that is the most common are Unit testing, Module testing, Integration testing, System testing, and Acceptance testing.

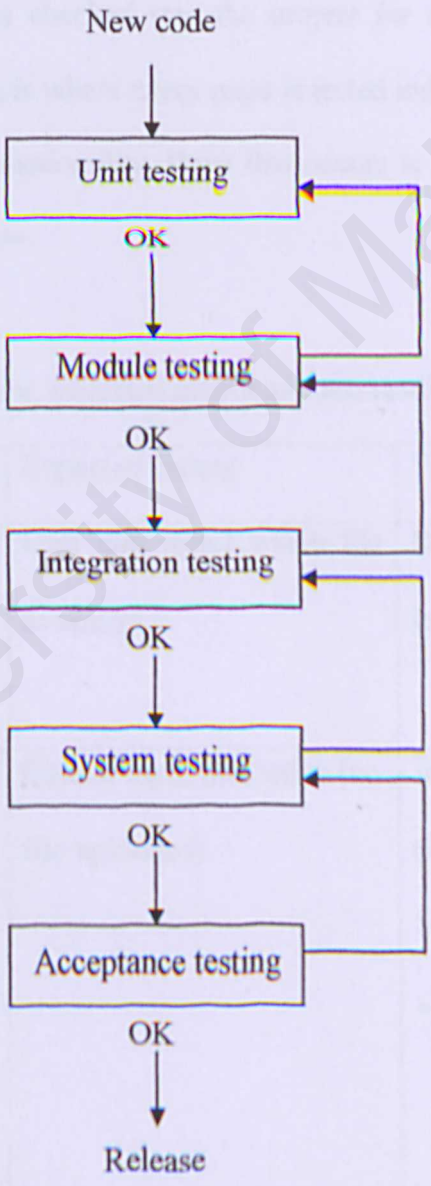


Figure 7.1: Testing strategies cycles

7.4.1 Unit Testing

A product unit is the smallest piece of code that can be independently tested. It is tested to make sure that the code works correctly. From an object-oriented programming perspective, classes, properties, method and events are all individual units. A unit should pass its unit test before it is checked into the project for integration. In the Image Retrieval System, unit testing is where every page is tested individually and Web Forms elements are checked their functionality. Error that occurs is fixed so that it would not affect other pages in the system.

Table 7.1: List of test procedure, expected output and test result / analysis in unit testing

No.	Test Procedure	Expected Output	Test Result / Analysis
1.	Click on the 'Browse' button of the file field.	User can select which file to upload.	Browse all file in any location in the computer.
2.	Button upload clicked.	Cannot input null value (no file uploaded).	Input any file into database (null or not null). Solution: • Change the code behind to restrict user to upload null file.

		Only input .gif file.	• Change the code behind to restrict user to upload only .gif image file. Provide error message each time other file type uploaded.
4.	Dropdown list to 'select image shape'.	Cannot input null value(-)	Done • Message will be prompt if null value is uploaded.
5	Delete	Delete file from database	Done using SQL Statement in the code-behind.
6	Sobel (edge detection)	Only .gif file type which are have transparent background and is a basic shape image can be detected.	Done using function written in the code-behind (VB.NET)
7	Reduce Chain Code	Convert image edges to image data.	Coding error Solution: • Try to fix the error
8	Weighted Levensthein Distance	Match the similar image	Coding error • Try to fix the error

7.4.2 Module testing

A module is a collection of dependent components. A module encapsulates all of these related components. Module testing enables each module to be tested independently. This testing will ensure that the module calling sequence in this project is systematic.

In module testing, two or more units in which either unit that use output data from or provide input data for another unit were tested in collection. These units have related characteristics to perform a common goal or function.

In the Image Retrieval System for Web, modules are divided into 4 main modules:

i. Sobel Edge Detection
ii. Reduced Chain Code
iii. Weighted Levenshtein Distance
iv. User Interface

Table 7.2: List of test procedure, expected output and test result / analysis in unit testing

No.	Test Procedure	Expected Output	Test Result / Analysis
1.	Sobel Edge Detection Module • Upload image .gif file type	Image with edge detected	Done

	• Image saves in database • Edge of this image is detected using this module		
2.	Reduce Chain Code (RDC) Module • Convert image edges to image data	Save the image data (0, 1, 2,3) in the database	• Programming error
3.	Weighted Levensthein Distance (WLD) Module • Match the similar image	Compare similar string (image data) in the database and return similar images	• Programming error • Have to implement RDC Module to make it a success
4.	• User Interface	Retrieve images through user query	• User query is performed by clicking on shapes to retrieve images with similar shape.

7.4.3 Integration test

The integration testing is to make sure that all units work together and compiled without error. At this point, a compilation error in any of the pages in the Image Retrieval System can keep the integration testing to move forward.

The most common build problem occurs when one component tries to use another component that has not yet been written. This occurs with modular design because the components are often created out of sequence. For example, the main program is accessing a function that has only declaration of the function. When built, it will return no error but the overall function does not return the intended output.

For integration testing in this system, each page that contains function and link will be tested to make sure that it function well and linked correctly.

Table 7.3: Integration testing of user pages

User pages		
Menu (Links):	**Link To (pages):**	**Result:**
1) About Us	aboutUS.aspx	Done
2) About IRS	index.aspx	Done
3) Home	index.aspx	Done
4) Select Image Base on Shape :	Retrieve image from database	

• Circle • Square • Oval • Triangle • Rectangle • Polygon	• Circle.aspx • Square.aspx • Oval.aspx • Triangle.aspx • Rectangle.aspx • Polygon.aspx	• Done • Done • Done • Done • Done • Done

Table 7.4: Integration testing of administrator pages

Administrator Pages		
Menu :	**Link To:**	**Result:**
1) Login	simple_login.aspx	Done
2) About IRS	index.aspx	Done
3) Instruction	instruction.aspx	Done
4) Upload Image	• ImageView.aspx • imager.aspx (page that display the image)	• Done • Done
5) Sobel	• sobel.aspx • sobel_display.aspx (page that contains the sobel function)	Done
6) Reduce Chain Code	• rcc.aspx	• Failed!

7) Weighted Levensthein Distance	• wld.aspx	• Failed!
8) Delete	• delete.aspx	Done
9) Select Image Base on Shape : • Circle • Square • Oval • Triangle • Rectangle • Polygon	Retrieve image from database • circle.aspx • square.aspx • oval.aspx • triangle.aspx • rectangle.aspx • polygon.aspx	 • Done • Done • Done • Done • Done • Done

7.4.4 System testing

The fact that the last code artifact has been integrated successfully into the system does not mean that the task of the developers is complete. System testing is done to ensure that the system fulfills the user requirements. The objective of this testing is to ensure that the system does what the users want it to do.

This testing procedure involves three kinds of system testing: functional testing, performance testing and acceptance testing.

7.4.4.1 Functional Testing

In functional testing [Howden, 1987], each item of funtionallity or function implemented in the code artifact is identified. After determining all the functions of a code artifact, test data are devised to test each function seperately.

7.4.4.2 Performance Testing

Performance test will be done after the functional testing. It will be when the system functions have been working as planned. Performance tests that will be conducted for the system are:

1. **Compatibility tests**
   - To test the interface functions

2. **Security tests**
   - To test the security of the application. In the Image Retrieval System, login page for the admin is tested to make sure that other authorized user can not enter into the admin system

3. **Recovery Tests**
   - To test the response when faults or loss of data, power, device or services occurs.

4. **Human Factors Tests**

   - To test the user interface of the system whether it fulfills the user interface requirement such as user friendly

5. **Browser independence tests**

   - To test the ability of the application to perform correctly to the specific web browser (Internet Explorer)

7.4.4.3 Acceptance Testing

The purpose of acceptance testing is for the client to determine whether this system satisfies its specifications as claimed by the developer. Acceptance testing for this system is done by user other than the developer. Feedback from the user is taken into account. Any changes regarding the feedback that can be made will be done or it will be on future planning for system enhancement plan.

7.5 Chapter Summary

This chapter briefly explains about the faults that can occur when writing a program code, planning a testing and types of tests carried out during the testing phase.

When writing code, faults that usually happen are algorithmic and syntax faults. Both of this faults must be detected and eliminate immediately to avoid any redundant error to other codes written.

CHAPTER 8:

SYSTEM EVALUATION

CHAPTER 8: System Evaluation

8.1 Introduction

This chapter will discuss the overall development process carried out and the implementation. Image Retrieval System for Web is planned and design on paper before implemented. The implementation of the system is where the design is translated in code. There are some part of the design is not implemented because of unexpected problem occurred but is overcome by some alternative solution made. Finally, the system is managed to be implemented and presented to the user.

8.2 Problem and solution

The problem that occurs during the system implementation will be discussed and solution will be provided.

8.2.1 Reduced Chain Code Module

This module is supposed to generate code for the image edges that had been detected in Module 1 (Edge Detection).

The developer had tried to write the function for this module by referencing to original programming code in Visual Basic (VB) language. But it is not a success because VB language provides different syntax compared to VB.NET language.

To solve this problem, an alternative provided whereby the admin have to select the type of image shape uploaded into the system. This enables the images retrieved based on the shape of images from the database.

8.2.2 Weighted Levenshtein Distance Module

To retrieved images from the database, the Weighted Levenshtein Distance is a matching technique to compare between similar images and return images that have the same shape.

But this module also could not be implemented because it depends on the implementation of module of Reduced Chain Code.

As a solution for this unimplemented module, images are then retrieved based on the shape selected by the admin. Similar shape image is return to the user by comparing the type of shape of the images through SQL Statement.

8.2.3 Error in uploading the type of shape

The task for selecting type of shape for images will be done by the administrator. With that, the admin have the probability to upload wrong type of shape for the images.

To solve this problem, the admin will have to be very careful when selecting the image shape.

8.3 System Evaluation by User

This system had been tested by users other than the developers. The testing leads to evaluation by users. Comments had been taken into account and will be used as future planning.

Table 8.1: User comment and action by developer

Comment	**Action**
1. Automatic matching with images in database based on shape detected	Future Planning: - Reduced Chain Code (RDC) to automatically generate code for image edge - Weighted Levenshtein Distance to match images based on comparing between similar strings from RDC.
2. Color based detection	Future Planning: Detect the edges and also the color of images uploaded into the system.
3. Enhance system for more advance use	Future planning: If the RDC, WLD and color based detection had been implemented, maybe the system can be enhanced for more critical application such as security and medical.

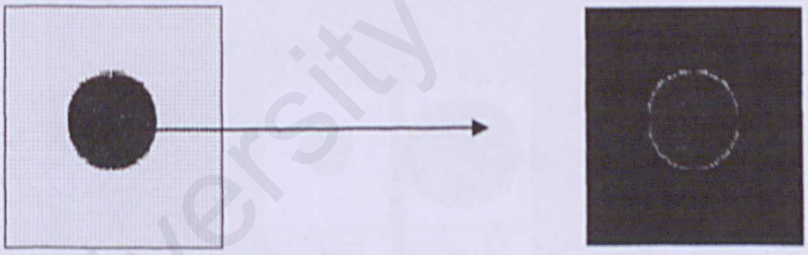
8.4 System strength

8.4.1 Web-based system

The Image Retrieval System for Web is a web-based system. With that, once this system is published in the internet, it can be access globally, 24 X 7. This means that a since the Image Retrieval System for Web is a web-based system it can be easily accessed by anyone around the world at any time.

8.4.2 Module of Edge Detection

The edges of the images in the database can be detected in the Image Retrieval System for Web. This module is important in order to know the shapes of the images.



Original image Image after edge detected

Figure 8.1: Example of edge detection

8.4.3 User friendly interface

The system developed have a user friendly interface to attract the user to use the system and for the ease of use. The interface is divided for the use of two types of user: User and Administrator.

The function menu is at the left side of the page and all the links targeted to the main frame to make is easier for the user to surf through the system without the hassle to go to new page for every link clicked.

In the example below shows that when the link circle is clicked, page circle will be display in the main frame.

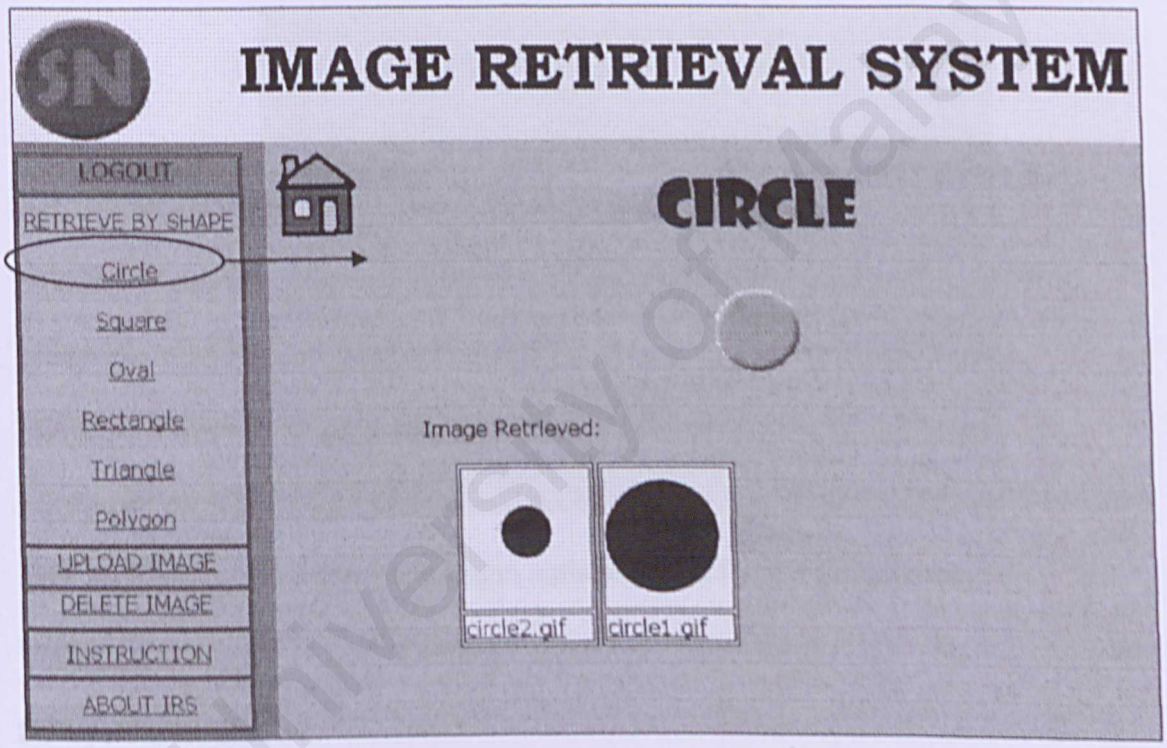


Figure 8.2: Example of user friendly interface

8.4.4 Security

In the Image Retrieval System for Web, the system had a security embedded. This is to enable the administrator to manage the system and also to prevent other user to

enter the secure function in the system such as upload and delete image from the database.

In the login authentication page, admin will have to enter the correct Admin ID and Password. If invalid Admin ID and password is entered then the admin could not enter into the admin page of the system.

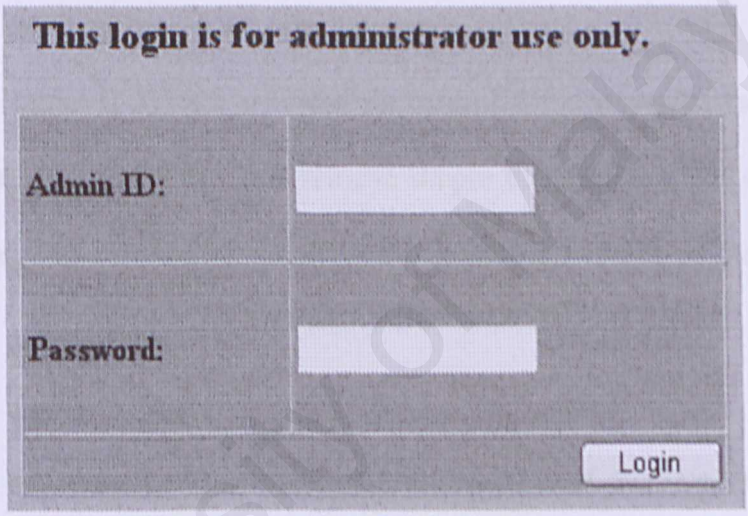


Figure 8.3: Login authentication for the administrator

8.5 System limitation

8.5.1 Web server: IIS

The Image Retrieval System for Web can be run only by using a specific web server: Internet Information Services (IIS) whereby this browser supports the ASP.NET language.

8.5.2 Images with file type .gif

The Image Retrieval System for Web can only upload .gif type of image. This is because VB.NET programming language have a specific function that can only do process on .gif type of image.

8.6 Future planning

The Image Retrieval System for Web implemented has provided the user with only basic type of retrieval. The user can retrieve shapes based on the type of shape set by the administrator. The function provided for the admin only to upload the image and detect the edges.

For future planning, both of the modules, Module of Reduced Chain Code (RDC) and Module of Weighted Levenshtein Distance (WLD) will be implemented in the system. The usage of the Edge detection, Coding using RDC and matching technique (WLD) will also be upgraded not only for the admin use but also for normal user of the system.

If all of the modules can be run smoothly, the use of the Image Retrieval System for Web can also be enhanced into the use for more effective and advanced application such as in Security and Medical perspective.

8.7 Experienced gained

The experiences gained after developing the Image Retrieval System for Web are:

i. Learned how to set up platform for developing a web based system
   - Setting up SQL Server to create database and tables and user access to the database.
   - Configuring IIS to create virtual directory for the system and to make it as the default web browser.
   - Installing Microsoft Visual Studio.NET and setting up project for the system development.

ii. Learned how to write program using new programming language of .NET environment (ASP.NET and VB.NET)

iii. Manage to develop and run a web-based system for the use of retrieving images from database using shape.

8.8 Chapter Summary

This chapter is the final chapter in this thesis. It discusses about the problems occurred during the implementation, system strength and also overall system development The future planning is also included in this chapter to enhance the system for future use future. Experienced gained discuss what had been learned after developing the Image Retrieval System for Web.

8.9 Overall conclusion for project

Image Retrieval System for Web is developed as the final year project by two developers; Normalyna Abdul Malek and Suhaili Beeran Kutty. It takes two semesters to complete the overall project.

The first part of the project was done in the first semester. It is mostly about planning and designing the system. The chapters in the first part of the project consist of Introduction, Literature Review, Methodology, System Analysis and System Design.

There was no major problem occur in the first part of the project. The only important aspect involves in the first part is the effort of finding the right resources to include as references to the project.

While the second part of the project was developing and implementing the system. It takes about 3 months to finish the development.

The Image Retrieval System for Web is developed using the Microsoft Visual Studio.NET software. The coding used for programming are ASP.NET and VB.NET. ASP.NET is used to code the interface meanwhile VB.NET is to write the function for pages developed in the system.

Problem that occurred during this part was that it takes time to set up the platform for development environment. Another problem is learning while using new programming language to develop the system as this is the first time that both of the developers use this language to write program.

To know how far this project is a success, reviews had been made on the objectives stated in the beginning chapter. Most of the objectives had been reached after the implementation of the project. The objectives reached are:

- To characterize and retrieve images based on their shape
  - This had been achieved because user can retrieve images based on their shapes by using the system
- To make it easier for the user to search for similar images in the database using input image or menu-driven query
  - Similar images can be retrieved using menu-driven query. In the system, images can be retrieved by clicking on the menu provided at the left side of the page.
- To provide a user friendly interface for this web-based system for the ease of use
  - The system provides a use friendly interface that is easy to use for both normal use and administrator.

In a nutshell, this system is a success because many objectives had been achieved and the system is implemented successfully. For the future plan, it is hope that this system able to use the RDC and WLD module to be implemented in the system and this system can be use for a more critical application such as Security and Medical application.

APPENDIXES

ImageView.aspx.vb

```
Imports System.Data
Imports System.Data.SqlClient
Imports System.Drawing
Imports System.IO
Imports System.Math

Public Class ImageView
    Inherits System.Web.UI.Page


    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

        If Not Page.IsPostBack Then
            BindData()
        End If

    End Sub


    Public Sub BindData()

        Dim sqlConnect As String
        sqlConnect = "workstation id=INA;packet size=4096;integrated security=SSPI;data
source=""(local)" & _
  """;persist security info=true;initial catalog=irs"
        Dim cn As New SqlConnection(sqlConnect)
        Dim sqlcmd As String
        sqlcmd = "Select * from ImageGallery order by ImageShape"
        Dim cmd As New SqlCommand(sqlcmd, cn)
        Dim da As New SqlDataAdapter(cmd)
        Dim dt As New DataTable("ImageGallery")
        da.Fill(dt)
        DataList1.DataSource = dt
        DataBind()

    End Sub

    Private Sub butUpload_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles butUpload.Click

        Dim imageStream As Stream = UploadFile.PostedFile.InputStream 'file uploaded
stream
        Dim imageLength As Integer = UploadFile.PostedFile.ContentLength
        Dim imageType As String = UploadFile.PostedFile.ContentType
```

```
    Dim imageshape As String
    Dim imageName As String

    imageName = UploadFile.PostedFile.FileName
    imageName = Path.GetFileName(UploadFile.PostedFile.FileName) 'get the name
of the file that is posted

    Dim mStream As New MemoryStream
    Dim imageData(1024) As Byte
    Dim count As Integer = imageData.Length

    count = imageStream.Read(imageData, 0, imageData.Length)
    Do While count > 0
      mStream.Write(imageData, 0, count)
      count = imageStream.Read(imageData, 0, imageData.Length)
    Loop

    If select1.SelectedIndex = (0) Then
      lblstatus.Visible = True
      lblstatus.Text = "Please choose the shape"
      Return

    ElseIf select1.SelectedIndex = (1) Then
      imageshape = "circle"
    ElseIf select1.SelectedIndex = (2) Then
      imageshape = "square"
    ElseIf select1.SelectedIndex = (3) Then
      imageshape = "oval"
    ElseIf select1.SelectedIndex = (4) Then
      imageshape = "rectangle"
    ElseIf select1.SelectedIndex = (5) Then
      imageshape = "triangle"
    ElseIf select1.SelectedIndex = (6) Then
      imageshape = "polygon"
    End If


    Dim sqlConnect As String
    sqlConnect = "workstation id=INA;packet size=4096;integrated security=SSPI;data
source=""(local)" & _
  """;persist security info=true;initial catalog=irs"
    Dim cn As New SqlConnection(sqlConnect)

    If (String.Compare(imageType, "image/gif", True) = 0) Then

      Dim sqlcmd As String
```

```
        sqlcmd = "Insert into ImageGallery(imageName, imageType, imageData,
imageshape)"
        sqlcmd &= "values(@imagename, @imagetype, @imagedata, @imageshape)"


        Dim cmd As New SqlCommand(sqlcmd, cn)
        cmd.Parameters.Add("@imageName", imageName)
        cmd.Parameters.Add("@imageType", imageType)
        cmd.Parameters.Add("@imageData", mStream.GetBuffer())
        cmd.Parameters.Add("@imageshape", imageshape)

        cn.Open()
        cmd.ExecuteNonQuery()
        cn.Close()
        BindData() 'bind data in data list
      Else
        Label4.Visible = True
        Label4.Text = "Input only image.gif"


      End If

    End Sub

End Class
```

imager2.aspx.vb

```
Public Class imager2
    Inherits System.Web.UI.Page

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim myid As String
        myid = Request("ID").ToString
        Image1.ImageUrl = "imager.aspx?source=db&id=" & myid
        HyperLink1.NavigateUrl = "sobel_display.aspx?id=" & myid 'Put user code to
initialize the page here
    End Sub

End Class
```

imager.aspx.vb

```
Imports System
Imports System.Drawing
Imports System.Data.SqlClient
Imports System.IO
Imports System.Math

Public Class Imager
   Inherits System.Web.UI.Page


   Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
      Dim strsql As String

      Dim ImageID As String
      If Request("ID") Is Nothing Then
        Return
      Else
        ImageID = Request("ID").ToString()
      End If
      strsql = "SELECT imageData, imageShape, imagename  FROM ImageGallery
 WHERE ImageID=" & ImageID

      Dim connstr As String = "workstation id=INA;packet size=4096;integrated
security=SSPI;data source=""(local)" & _
  """;persist security info=true;initial catalog=irs"
      Dim cnn As New SqlConnection(connstr)
      Dim cmd As New SqlCommand(strsql, cnn)
      cnn.Open()
      Dim dr As SqlDataReader = cmd.ExecuteReader
      dr.Read()

      'Response.ContentType = "image/jpeg"
      Response.BinaryWrite(dr.GetValue(0))
      cnn.Close()
      cnn.Dispose()
      Response.End()
   End Sub

End Class
```

sobel.aspx.vb

```
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.Math
Imports System.IO
Imports System.Data.SqlClient


Public Class sobel
   Inherits System.Web.UI.Page


   Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
      Dim i As Image
      Dim strsql As String

      Dim ImageID As String
      If Request("ID") Is Nothing Then
         Return
      Else
         ImageID = Request("ID").ToString()
      End If

      Dim sqlConnect As String
      sqlConnect = "workstation id=INA;packet size=4096;integrated security=SSPI;data
source=""(local)" & _
  """;persist security info=true;initial catalog=irs"
      Dim cn As New SqlConnection(sqlConnect)
      Dim sqlCmd As String
      sqlCmd = "Select imageType, imageData from ImageGallery where"
      sqlCmd &= " ImageID = @ImageID"
      Dim cmd As New SqlCommand(sqlCmd, cn)
      cmd.Parameters.Add("@imageID", ImageID)

      cn.Open()

      Dim dr As SqlDataReader = cmd.ExecuteReader()
      If Not dr.Read() Then
         Response.Write("image not found")
         Return
      End If

      Response.ContentType = dr("imageType".ToString())

      Dim mStream As MemoryStream
```

```
        Dim byteData() As Byte
        byteData = dr("imageData")
        mStream = New MemoryStream(byteData)

        i = Image.FromStream(mStream)
        Dim b As New Bitmap(i)
        b = graypixel(b, b.Width, b.Height)

        ' Set the content type
        Response.ContentType = "image/gif"

        ' send the image to the viewer
        b.Save(Response.OutputStream, ImageFormat.Gif)

        ' tidy up
        b.Dispose()

    End Sub


    Function graypixel(ByVal b As Bitmap, ByVal x As Integer, ByVal y As Integer)
        Dim addRGB As Integer
        Dim tmpColor
        addRGB = 0

        ' for each pixel
        For y = 3 To b.Height - 1
            For x = 3 To b.Width - 1

                ' average out the rgb
                tmpColor = b.GetPixel(x, y)
                addRGB = tmpColor.r
                addRGB += tmpColor.g
                addRGB += tmpColor.b
                addRGB = addRGB / 3


                ' then Set the values
                b.SetPixel(x, y, Color.FromArgb(addRGB, addRGB, addRGB))

            Next
        Next

        Return b
        b.Dispose()
    End Function
End Class
```

sobel_display.aspx.vb

```
Imports System.Drawing
Imports System.Drawing.Imaging
Imports System.IO
Imports System.Data.SqlClient

Public Class imagerNext
    Inherits System.Web.UI.Page


    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim myid As String
        myid = Request("ID").ToString
        Image1.ImageUrl = "sobel.aspx?id=" & myid

    End Sub

End Class
```

delete_page.aspx.vb

```
Imports System.Data.SqlClient

Public Class delete_page
    Inherits System.Web.UI.Page

    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim strsql As String

        Dim ImageID As String
        If Request("ID") is Nothing Then
            Return
        Else
            ImageID = Request("ID").ToString()
        End If

        strsql = "DELETE FROM ImageGallery WHERE imageID=" & ImageID

        Dim connstr As String = "workstation id=INA;packet size=4096;integrated
security=SSPI;data source=""(local)" & _
    """;persist security info=true;initial catalog=irs"
        Dim cnn As New SqlConnection(connstr)
        Dim cmd As New SqlCommand(strsql, cnn)
        cnn.Open()
        Dim dr As SqlDataReader = cmd.ExecuteReader

        cnn.Close()
        cnn.Dispose()

        Response.Redirect("delete.aspx") 'Put user code to initialize the page here
    End Sub

End Class
```

delete.aspx.vb

```
Imports System.Data.SqlClient

Public Class delete1
    Inherits System.Web.UI.Page


    Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        SqlDataAdapter1.Fill(DataSet91)
        DataList1.DataBind() 'Put user code to initialize the page here
    End Sub


End Class
```

circles2.aspx.vb

```
Imports System.Data.SqlClient
Imports System.Data.OleDb
Imports System.IO
Public Class circles2
   Inherits System.Web.UI.Page

   Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
      Dim strsql As String

      Dim ImageID As String
      If Request("ID") Is Nothing Then
         Return
      Else
         ImageID = Request("ID").ToString()
      End If

      If Not Request("Imageshape") Is Nothing Then
         If (String.Compare( _
         Request("imageShape").ToString(), _
            "circle", True) = 0) Then

            strsql = "SELECT imageData, imageShape, imagename  FROM ImageGallery
WHERE (imageShape='circle') AND ImageID=" & ImageID
         End If
         Dim connstr As String = "workstation id=INA;packet size=4096;integrated
security=SSPI;data source=""(local)" & _
  """;persist security info=true;initial catalog=irs"
         Dim cnn As New SqlConnection(connstr)
         Dim cmd As New SqlCommand(strsql, cnn)
         cnn.Open()
         Dim dr As SqlDataReader = cmd.ExecuteReader
         dr.Read()

         'Response.ContentType = "image/jpeg"
         Response.BinaryWrite(dr.GetValue(0))
         cnn.Close()
         cnn.Dispose()
         Response.End()
      End If

   End Sub

End Class
```

circles.aspx.vb

```
Imports System.Data.SqlClient
Imports System.Data.OleDb
Imports System.IO
Public Class circles
   Inherits System.Web.UI.Page

   Private Sub Page_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load

      SqlDataAdapter1.Fill(DataSet11)
      DataList1.DataBind()

   End Sub

End Class
```

REFERENCES

REFERENCES

Azimah binti Razali. (2002). *Image Edge Detection System.* Bachelor Thesis. University of Malaya.

Bouet, M., Khenchaf, A., and Briand, H. (1999). Introduction & Shape Representation. *Shape Representation for Image Retrieval*. 1-2.

Cortelazzo, G., Mian, G.A., Vezzi, G. and Zamperoni, P. (1994). Pattern Recognition. *Trademark Shapes Description by string-matching techniques.* **27(8).** 1005-1017.

Mehtre, B.M., Kankanhalli, M.S., and Lee, W.F. (1995-1996). *Shape Measures for Content Based Image Retrieval: A Comparison.*

Muhamad Najib bin Zakaria. (2002) *Sistem Imej Kepada Data (SIKD)*. Bachelor Thesis. University of Malaya.

Norjihan Abdul Ghani. Laporan Teknikal. Universiti Kebangsaan Malaysia.

Stair, R. M., Reynolds G. W. (2001). *Principles of information system: a managerial approach*. 5th ed. Course Technology.

Whitten, J.L., Bentley, L.D., and Dittman K.C. (2004). *System Analysis & Design Methods*. 6th ed. McGraw-Hill Companies.

Internet Reference

(URL-http://ausweb.scu.edu.au), 28/06/2004.

(URL-http://www.cee.hw.wc.uk.org), 08/07/2004.

(URL-http://www.generation5.org), 12/07/2004.

(URL-http://www.merriampark.com), 27/07/2004.

(URL-http://elib.cs.berkeley.edu), 04/08/2004.

(URL-http://amazon.ece.utexas.edu/~qasim), 04/08/2004.

(URL-http://elib.cs.berkeley.edu/photos/blobworld/), 22/09/2004

(URL-http://homepages.inf.ed.ac.uk/rbf/HIPR2/canny.htm), 27/09/2004.

(URL-http://www.planet-source-code.com/vb/scripts/ShowCode.asp?txtCodeId=11631&lngWId=1), 25/01/2005.

(URL-http://www.pages.drexel.edu/~weg22/edge.html), 25/01/2005.

(URL-http://www.leadtools.net/sdk/sdk%20help/commandservice/Sobel.htm), 25/01/2005.

(URL-http://marathon.csee.usf.edu/edge/edge_detection.html), 25/01/2005.

(URL-http://www.dotnetbips.com/displayarticle.aspx?id=101), 25/01/2005.

(URL-http://www.sitepoint.com/forums/showthread.php?t=38728), 25/01/2005.

(URL-http://www.codeproject.com/asp/Display_Images_from_Data.asp), 25/01/2005.

(URL-http://odetocode.com/Articles/163.aspx), 25/01/2005.

(URL-http://www.programmingtalk.com/archive/index.php/t-1981.html), 25/01/2005.

(URL-http://www.ittc.ku.edu/~jgauch/research/kuim/html/06.00.html), 27/01/2005.

(URL-http://www.catenary.com/howto/diagedge.html), 27/01/2005.

(URL-http://www.catenary.com/howto/diagedge.html), 27/01/2005.

(URL-http://www.ragingsmurf.com/code.aspx?key=0ZUJ81YB91), 03/02/2005.

(URL-
http://subscribe.ru/archive/comp.soft.prog.vbnewsadvices/200302/28070913.html),
03/02/2005.

(URL-http://www.squirreldome.com/Examples.htm), 07/02/2005.

(URL-http://www.microsoft.com/resources/documentation/sql/2000/all/reskit/en-
us/part3/c1161.mspx), 07/02/2005.

(URL-http://www.dotnetindex.com/articles/1/?SESSION=877303214&REF=),
07/02/2005.

USER MANUAL

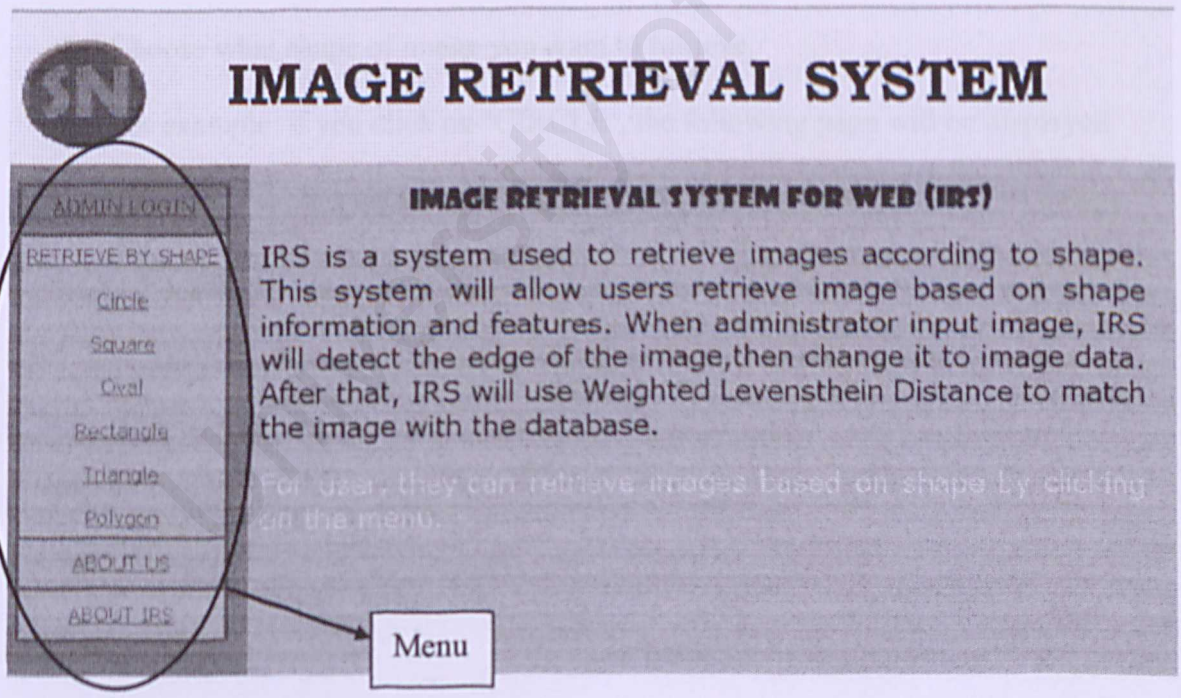
IMAGE RETRIEVAL SYSTEM USER MANUAL

Introduction

Image Retrieval System for Web is developed for user to retrieve images according to their basic shapes. This system is divided into user and administrator use.

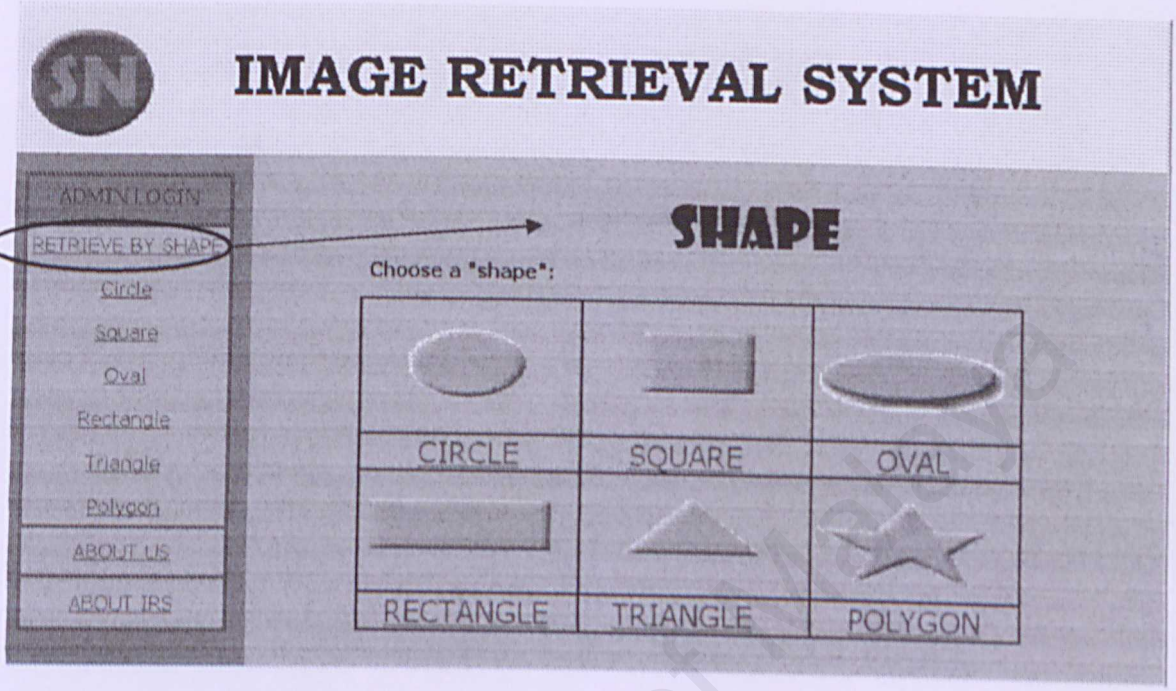
USER

1. When you open the Image Retrieval System for Web (IRS), the main page will be the 'About IRS' page.



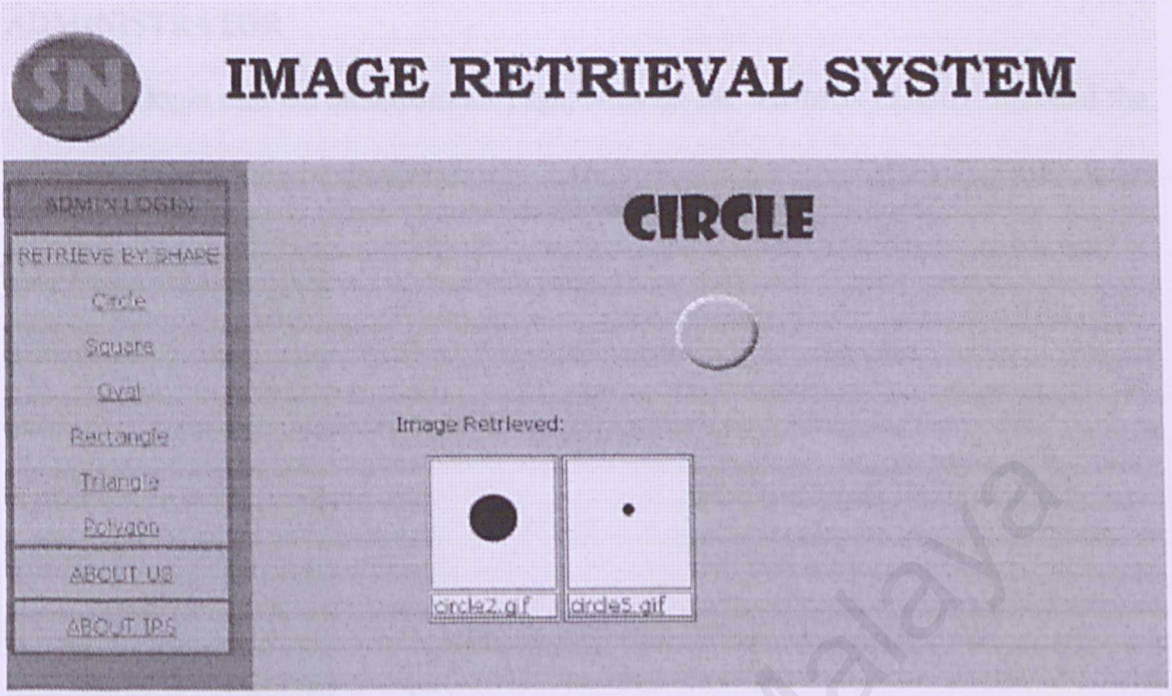
2. All the links to pages in the IRS are in the Menu at the left side of the page.

3. To retrieve images based on their shape, click on the link 'RETRIEVE BY SHAPE' link and the page 'SHAPE' will appear at the main page.

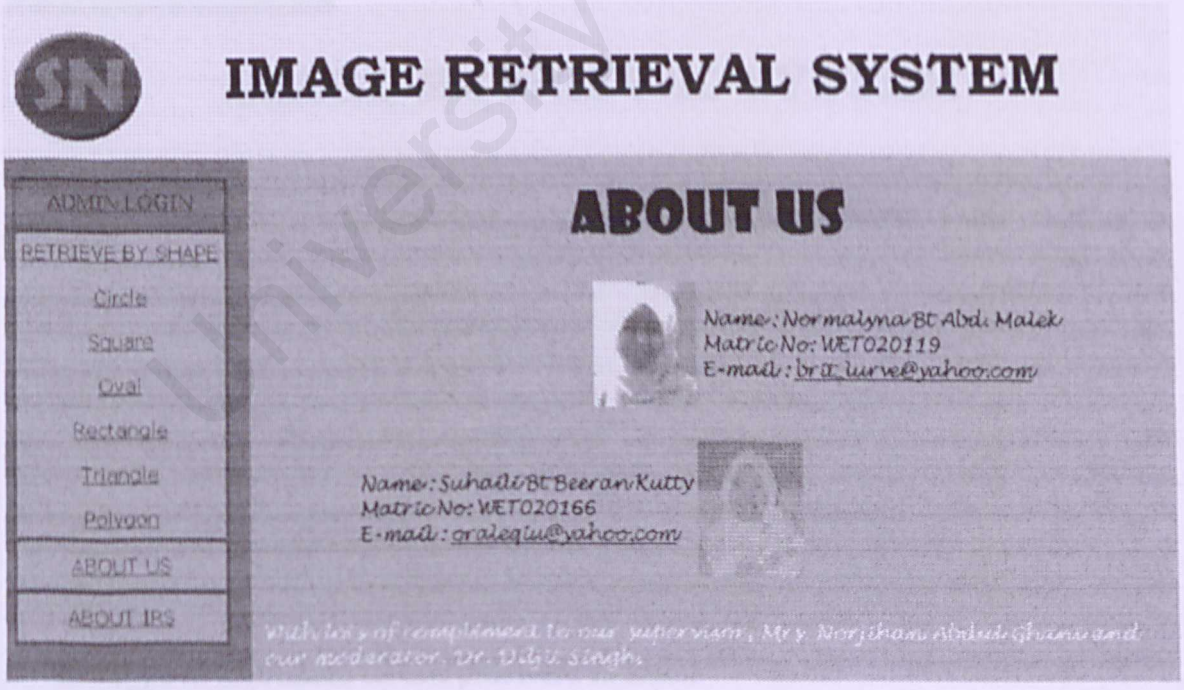


4. Choose what shape of image you want to retrieve.
5. For example, if you click on 'CIRCLE', the following page will be displayed.

- You can also click on the link 'Circle' in the Menu to retrieve the similar page.



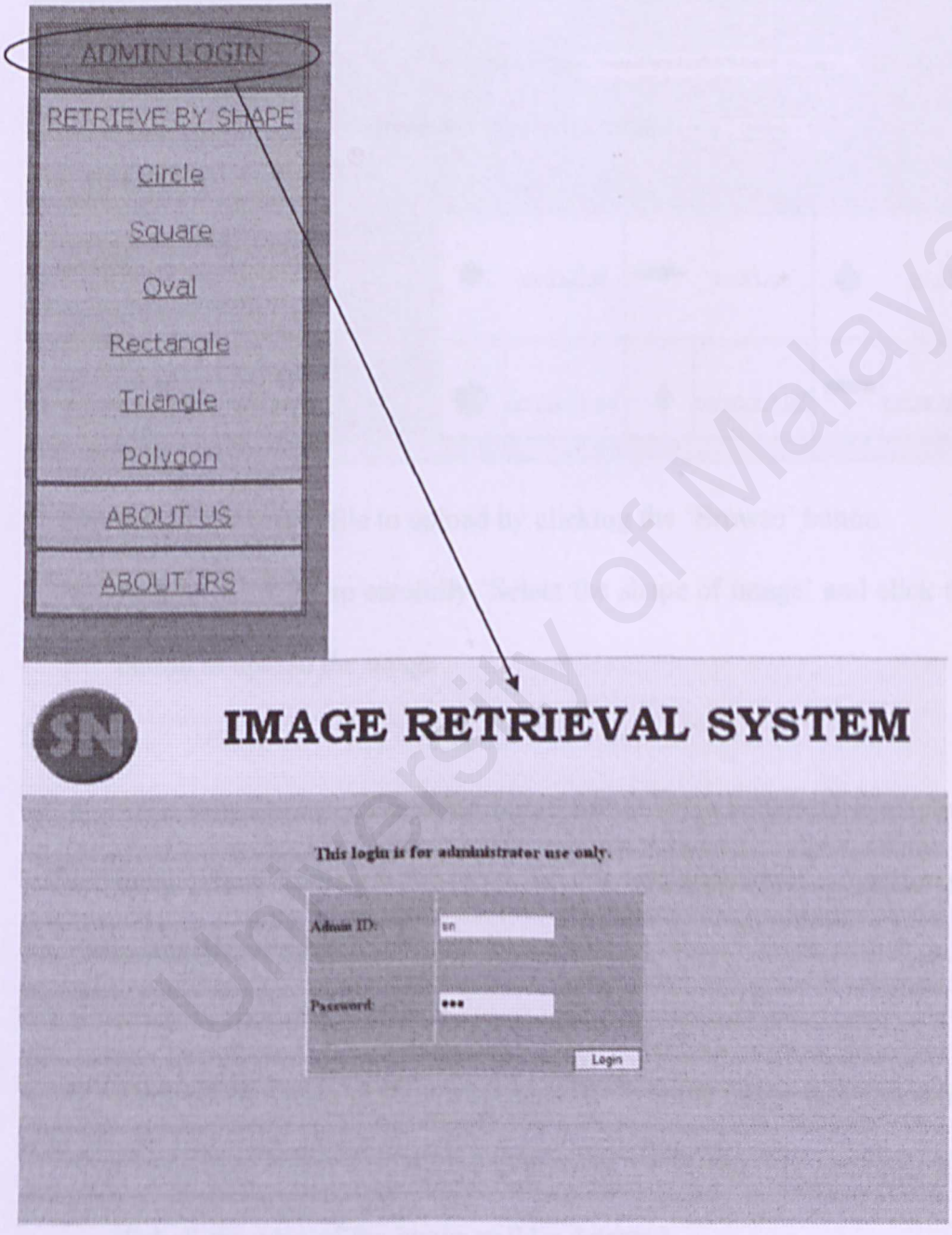
6. To know more about the developer, click on the 'ABOUT US' link.



7. The link 'ABOUT IRS' will direct you to the main page.

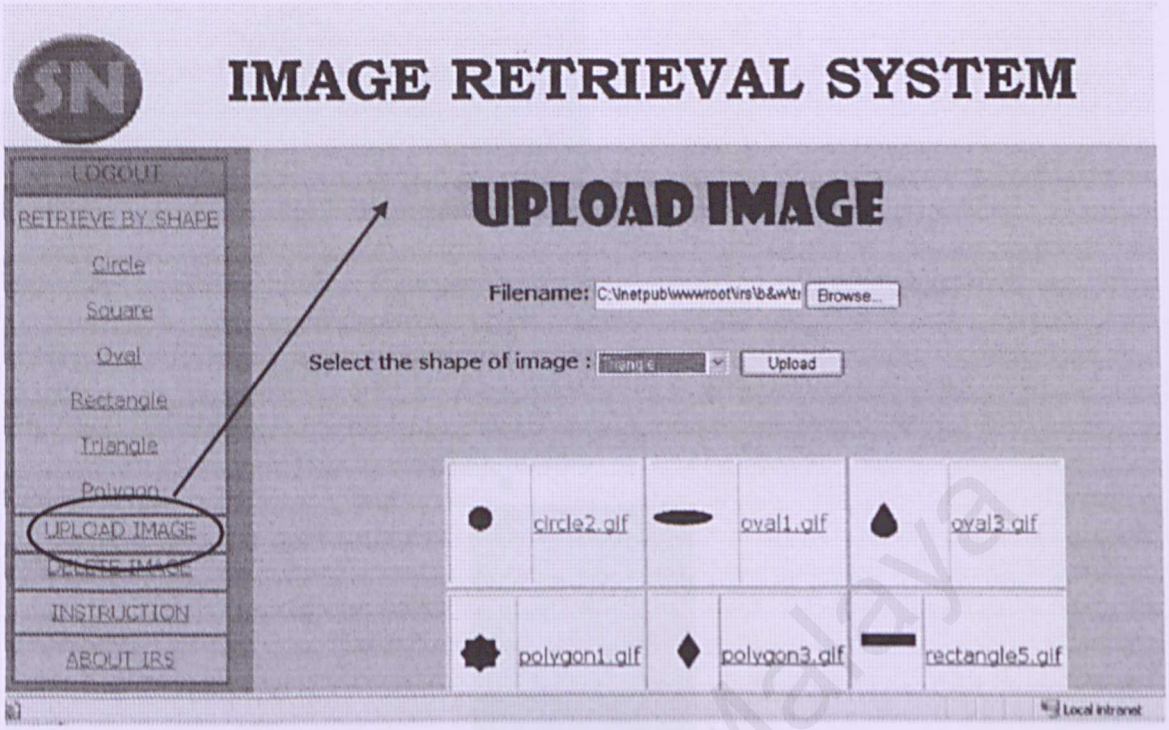
ADMINISTRATOR

1. To login into the administrator page, click on the 'ADMIN LOGIN' link and the login page will be displayed.



2. Type in the Admin ID and password to login.

3. To upload images in the database, click on the 'UPLOAD IMAGE' link.

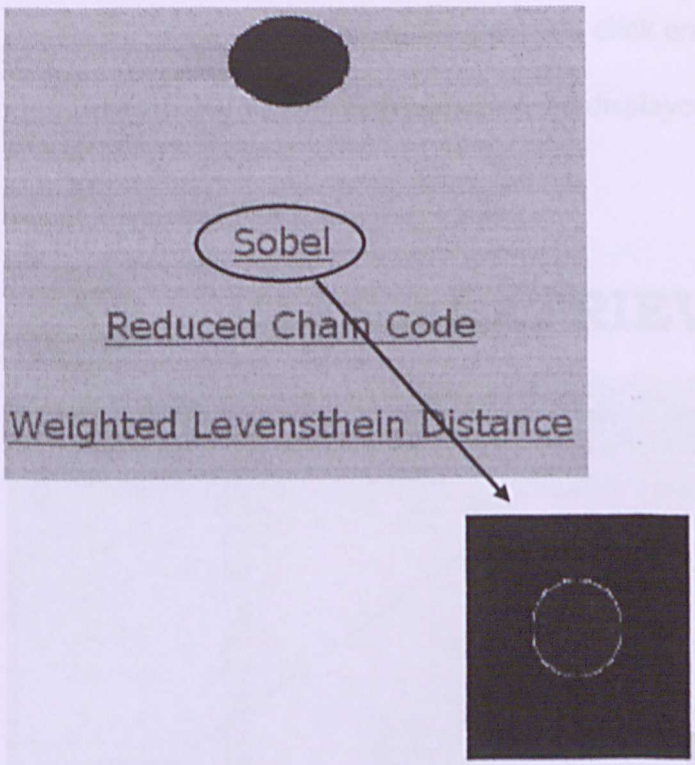


4. Select any image file to upload by clicking the 'Browse' button.
5. Admin will have to carefully 'Select the shape of image' and click the 'Upload' button to upload the image.

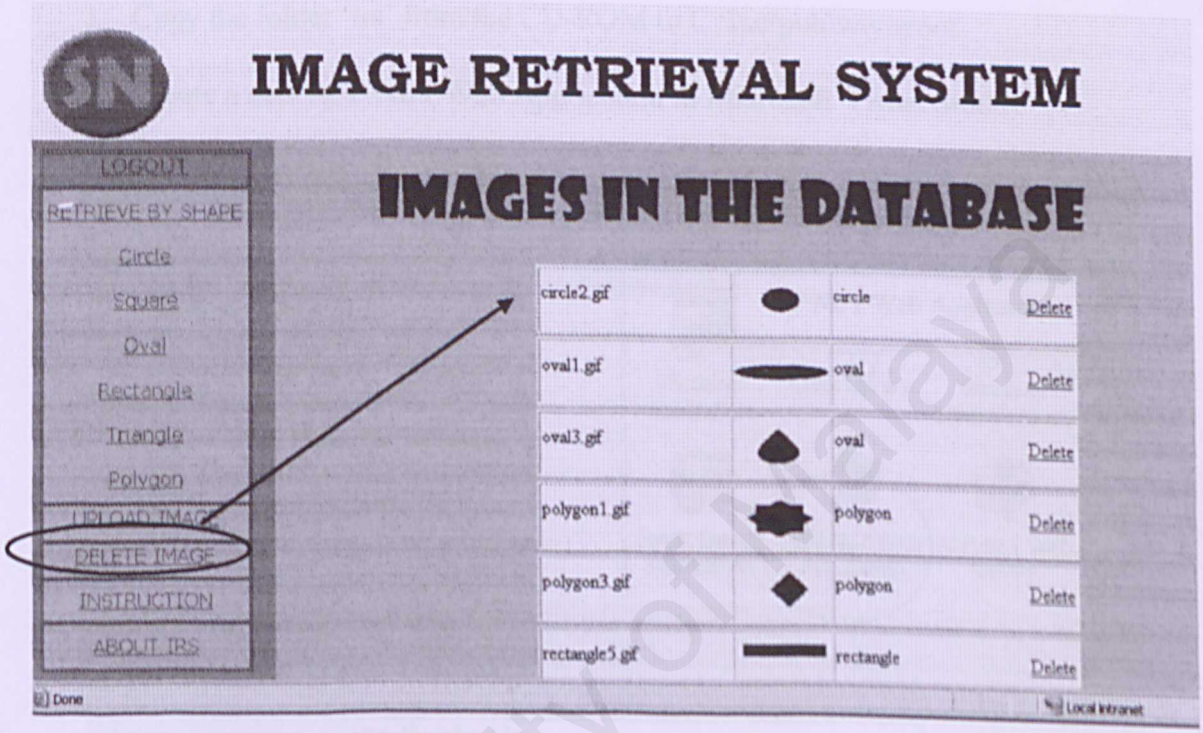
NOTES:

- Error message will remind the administrator if **No Filename** is entered.
- Only image with .gif file type can be uploaded in the system or else error message will be displayed.

6. Click on the image or the image name to view the image in a new window.
7. Choose any function provided to process the image. For example, if you choose 'Sobel' the edge of the image will be detected.



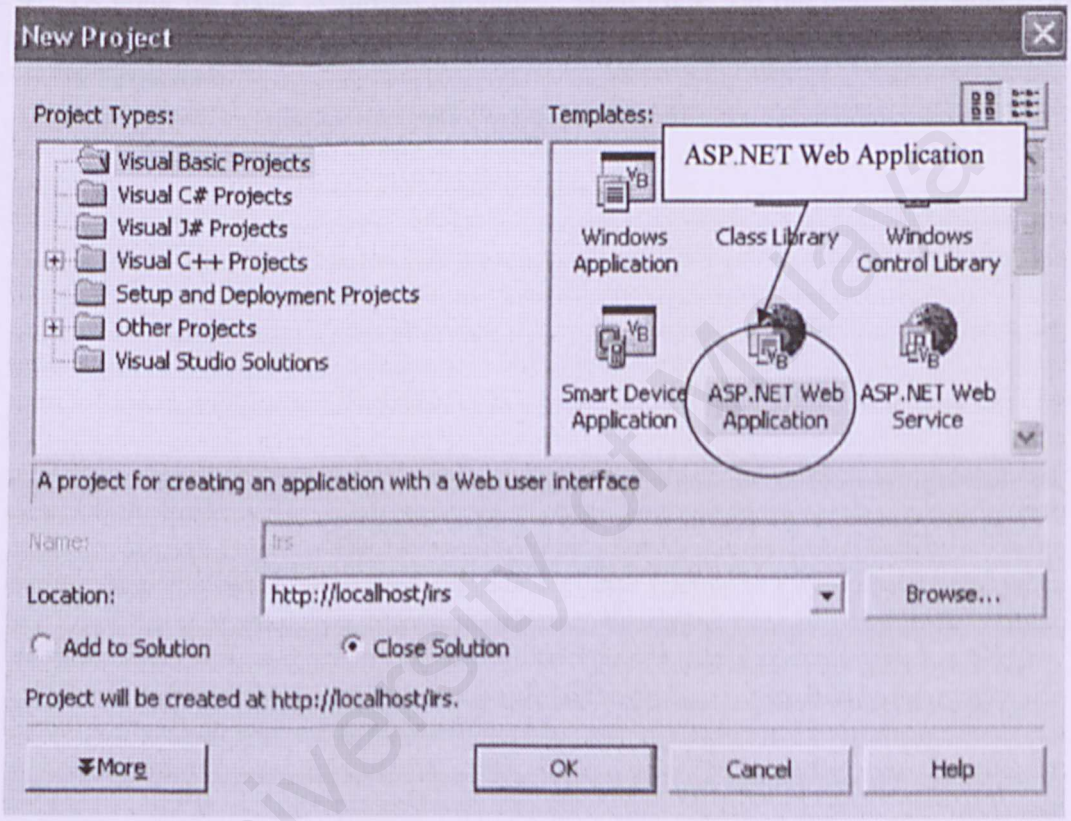
8. To delete any images in the database, click on the link 'DELETE IMAGE' in the Menu. All images in the database are displayed in a list.



9. Click on the 'Delete' link to delete the selected image.
10. The 'INSTRUCTION' link will direct the admin to a page that provide instruction on how to 'Upload image' and 'Delete image'.
11. To logout, click on the 'LOGOUT' link and you will logout from the administrator page. You will be directed back to the User Page.

To setup the Image Retrieval System for Web project in Microsoft Visual Studio.NET

1. Copy the folder 'irs' from the CD-ROM to C:\Inetpub\wwwroot.
2. Create a new ASP.NET Web Application in Microsoft Visual Studio.



3. Select Location: http://localhost/irs and click OK.
4. All the pages in the folder 'irs' will be included in the 'irs' project in Microsoft Visual Studio.NET environment.
5. In the Server Explorer, Add Connection at the Data Connection. This connection is to connect with the database in the SQL Server.
6. To enable pages to run smoothly, the Connection String in each page must be change because it Connection String at each computer is different. For example, the connection string made in the Computer at Faculty Lab:

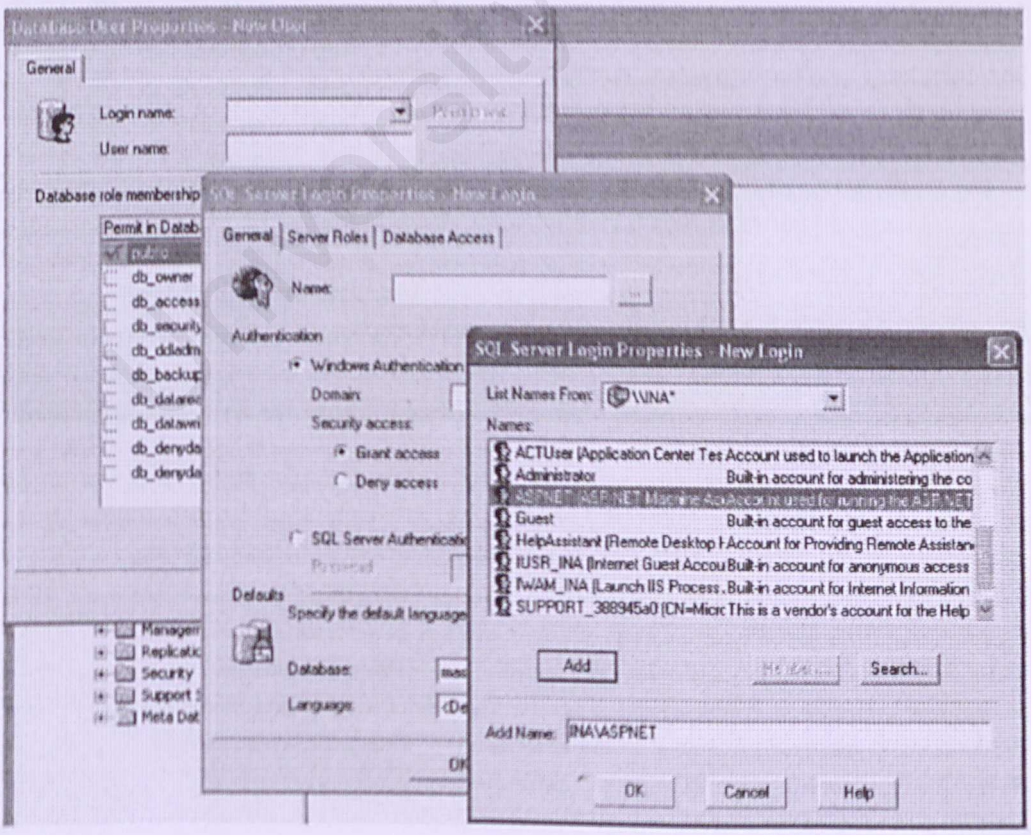
"workstation id=ANX736;packet size=4096;integrated security=SSPI;data source=""ANX7" & _

"36\LOCALHOST"";persist security info=False;initial catalog=irs"

7. To view the page in the design view in the 'irs' project, you just have to 'double click' on the selected file.
8. To view the code-behind, 'double click' on the design view of the page.
9. To view the page in virtual directory, 'right click' on the page and select 'View in Browser'.

To setup the IRS database in the SQL Server

1. Create a new database in the SQL Server named 'irs'.
2. Copy irs_Data and irs_Log from the CD-ROM to C:\Program Files\Microsoft SQL Server\MSSQL\Data.
3. Replace the existing file in the location.
4. Next, go back to the 'irs' database in SQL Server.
5. Create a new user for the database 'irs'.
   - Right click on the 'Users'.
   - Select New Database User
   - Login name: <new>
   - Select 'Name' from the list provided. Add Name: ASP.NET



6. After that, Manage Permission for the user that had been created to permit the Administrator to SELECT, INSERT, UPDATE and DELETE of the table ImageGallery. This will enable the administrator to upload, view and delete images through the user interface of the Image Retrieval System for Web.