

Chapter One

Introduction

1.1 Overview

Computer networks have grown explosively in the last two decades. In the past, fewer number of people used networks, but this number has increased year by year until these days where the computer networks have become one of the most common way to communicate with the world.

The system implemented in this project provides a set of services to the user in signal processing field. The Matlab software used as the processing software platform because one of the most processing fields the Matlab it can do is the signal processing. The Matlab provides very strong and powerful platform tools to apply simple and complex signal processing algorithms on different types of data.

Depending on the Matlab and the networking it was decided that it is a good idea to work in these fields.

The title of the project is **Digital Signals Processing Center**, this title came from the idea of processing user data by a set of algorithms via the Internet.

The first step in the project is to find the way that can access and execute the Matlab files via the web, and this will be done by using Matlab Web Server.

The next step is to put a set of algorithm related to a digital signal processing in the web application to be accessed by the users of the web application. To make the project more powerful and easy to use it must have a good user interface to perform user needs.

The project has a set of image and sound algorithms to process one and two dimension signals, these algorithms are programmed by using the Matlab software.

To increase the application control and management a database part was added to the project, this part manages the user's accounts and control the access of the users and the usage of algorithms.

This project needs to use Matlab, PHP, MySQL, HTML, Apache web server and PostCast mail server as software platform to perform the aim of the project in a good and effective way.

1.2 System Objectives

- ? To give users the ability to access and execute complex algorithms.
- ? To access user file and upload it to the server to perform and apply the selected user algorithm on this file.
- ? To send the processing result to any where in the world via email.
- ? To create and control user's account in addition to manage the accessibility of the system.
- ? To save money, time and effort when performing tasks especially when the users haven't Matlab installed on their computers, and also when they want to use algorithms and they are not interested in knowing how to build their codes.
- ? To make the system flexible to enable the system owner to implement and operate his own algorithms in easy way.

The project initially works with the signal processing application and the project may be modified to give any services Matlab can do such as medical, financial, control application etc.

1.3 Project Benefits for the Project Team

Working on this project has given the work group the following benefits:

- ? Learning very strong software like Matlab.

- ? Learning and working with web programming languages such as active server pages specially PHP, in addition to use web application design programs such as Macromedia Dreamweaver MX.

- ? Working with great database system by using MySQL.

- ? Studying and using a set of digital signals processing algorithms such as wavelet de-noise, filtering, edge detection algorithms and others.

- ? Learning information about the file transfer using the web; additionally learn about dealing with emails.

1.4 Why Dynamic Web Documents?

It provide the ability to report current information in good way whenever a browser requests the information, the server runs an application that accesses the needed information and creates a document, the server then sends the document to the browser.

Dynamic document places the responsibility on the server, a browser uses the same interaction to obtain a dynamic page as retrieve a static pages. From the browser's view, dynamic pages are indistinguishable from the static page because both use HTML, a browser does not know whether the server extracted the page from a disk file or obtained it from dynamic page. [3][5]

1.5 Why PHP?

PHP (Personal Home Page) set its roots in 1995, when an independent software development contractor named *Rasmus Lerdorf* developed a Perl / CGI script that enabled him to know how many visitors were reading his online resume.[3][5][6]

PHP is about providing the programmer with the necessary tools to get the job done in a quick and efficient fashion. PHP embracing of this open source strategy results in great performance gains for users.

Five important characteristics make PHP's practical nature possible:

? **Familiarity:**

Programmers from many backgrounds will find themselves already accustomed to the PHP language. Many of the languages constructs are borrowed from C and Perl, and in many cases PHP code is almost indistinguishable from that found in the typical C or Pascal program. This minimizes the learning curve considerably. [6]

? **Simplicity:**

A PHP script can consist of 10,000 lines or one line: whatever you need to get the job done. There is no need to include libraries, special compilation directives. The PHP engine simply begins executing the code after the first any-thing escape sequence (<?) and continues until it passes the closing escape sequence (?>). If the code is syntactically correct, it will be executed exactly as it is displayed. [5]

? **Efficiency:**

Efficiency is an extremely important consideration for working in a multi-user environment such as the World Wide Web. PHP 4.0 introduced resource allocation and more pronounced support for object-oriented programming, mechanisms into session management features. Reference counting has also been added in the latest version, eliminating unnecessary memory allocations.[6]

? **Security:**

PHP provides developers and administrators with a flexible and efficient set of security safeguards. These safeguards can be divided into two frames of reference system level and application level. [3][6]

? **Flexibility:**

Because PHP is an embedded language, it is extremely flexible toward meeting the needs of the developer. Although PHP is generally touted as being used in solely with HTML, it can also be integrated alongside languages like JavaScript, conjunction WML, XML, and many others. Additionally, with most other mainstream wisely planned PHP applications can be easily expanded as needed.

Browser dependency is not an issue because PHP scripts are compiled entirely on the server side before being sent to the user. In fact, PHP scripts can be sent to just about any kind of device containing a browser, including cell phones, personal digital assistant (PDA) devices, pagers, laptops, not to

mention the traditional PC. People who want to develop shell-based applications can also execute traditional PHP from the command line. Since PHP contains no server specific code, users are not limited to a specific and perhaps unfamiliar Web server. Apache, Microsoft IIS, Netscape Enterprise Server, Stronghold, and Zeus are all fair game for PHP's server integration. [4][5][6]

For all these reason it was decided to work with PHP. Additionally to make the system compatible to work with the UNIX platform because the Matlab, PHP and apache web server are compatible with this platform by using simple configurations.

1.6 Why Matlab? [1][2]

MATLAB is a high-performance language for technical computing. It integrates computation, visualization and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation.

Typical uses include Math, computation algorithm development, data acquisition modeling, simulation, prototyping data analysis, exploration, visualization scientific and engineering graphics application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows the user to solve many technical computing problems, especially those with matrix and vector formulations.

MATLAB has evolved over a period of years with input from many users. In university environments, it is the standard instructional tool for introductory and advanced courses in mathematics, engineering and science. In industry, MATLAB is the tool of choice for high-productivity research, development and analysis.

MATLAB features a family of add-on application-specific solutions called toolboxes; toolboxes allow the user to learn and apply specialized technology.

1.7 Why MySQL? [7][8]

MySQL, the most popular open source SQL database, is developed, distributed and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers that build its business providing services around the MySQL database.

The main features of MySQL:

- ? Works on many different platforms.
- ? Uses GNU (General Public License), Automake, Autoconf, and Libtool for portability.
- ? APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl.
- ? Fully multi-threaded using kernel threads. This means it can easily use multiple CPUs if available.
- ? Transactional and non- transactional storage engines.
- ? A very fast thread- based memory allocation system.
- ? Support many data types (integers, string, float, double, char, varchar, text, blob, date, time, date, time).
- ? Fixed- length and variable- length records.
- ? A privilege and password system that is very flexible , secure and allows host - based verification. Passwords are secure because all password traffic is encrypted when you connect to a server.
- ? Handles large databases 60,000 tables and about 5,000,000,000 rows.

The project databases design section (4.2) records much information about this subject not only about MySQL but also about the system database.

1.8 Report Contents

This report shows all important points related to the project, this documentation helps the reader to get comprehensive view about the project.

We tried to present a documentation that can be understood easily by dividing the documentation to a set of subject related parts.

The report consists of a number of chapters. Each chapter discusses a subject related to the project, and these chapters are sorted according to logical relations of subjects; through this document there are a lot of tables, graphics and charts that help in understanding the presented concepts.

Chapter one gives general information about the project, introduction about the project, system objectives, project benefits for the project teams and the system implementation tools and components.

Chapter two shows the system planning including the introduction about the system, the objectives of the system, the risk analysis, feasibility study, economical study and time schedule.

Chapter three discusses the requirement analysis which includes the functional requirements, requirement definition, requirement specification and discusses the non-functional requirements.

The system design founds in chapter four which consists of the system design model which includes the context, data flow diagram, structural chart and the data and control analysis, the second part of this chapter is the system database design.

Chapter five Implementation shows many ideas such as general view; prepares the platform of the system, Web site design and implementation.

Chapter six is the testing chapter that starts with an introduction, then testing scheduling, testing procedures and testing strategies.

Chapter seven is the conclusion and future work chapter which gives the conclusion about this project and gives the future work ideas.

At the end of the documentation there is a set of appendices that discusses in more detailed some subjects as needed to increase the reader understanding.

Chapter 2

System Planning

2.1 Introduction

This system is built on networks and Matlab based system, to take some of Matlab software capabilities and use it in network application.

Because the Matlab gives strong and powerful tools for signal processing it was decided to built a system and called it **Digital Signals Processing Center**, to give the users the ability to apply special processing algorithms on their files.

2.2 Objectives

It was decided to create a complete web application to provide easy and powerful platform to the users to process their own digital signal by applying special algorithms the system has and send back the processed result to the user.

The system users do not need to install the Matlab on their computers or even to buy it. They can simply access the system remotely via the Internet and choose the algorithm they want to work with. Only the server has the Matlab installed on it.

The user can choose any compatible file on his computer and apply suitable algorithm the system provide on this file, the system provides a number of image and sound algorithms.

In addition to that the user can send the processing result to any where in the world as E-mail.

The system gives the administrators the ability to control and access the algorithms, manage the user's accounts and also add new algorithm remotely.

2.3 Risk Analysis

This part shows some of the system risks, a lot of these risks can be reduced by more and more research about this system.

1. The web site may be attacked by professional hackers.
2. The loss of the user ID and password will prevent the user from using the system.
3. The bad files such as files having viruses cause a problem to the system.
4. Badly email information will cause sending the file to unknown destination at the routing stage of the result.
5. The heavy load on the server could cause a system low response problem.

2.4 Project Planning

2.4.1 Feasibility Study

2.4.1.1 Economical Study

Operating Hardware Costs:

Table 2.1 Operating Hardware Costs.

1	Microsoft compatible PC Pentium (Pentium 4 and up recommended).	1100\$ x 1
2	Network adapter card.	25\$ x 1
3	Network cables (5 meters).	2\$ x 5
	Total	1135\$

Operating Software Costs:

Table 2.2 Operating Software Costs.

1	Windows 2003 Web Server.	1000\$
2	Matlab 6.5.1	1500\$
3	Macromedia Dreamweaver.	400\$
4	My SQL	Free
5	Apache	Free
6	PHP	Free
	Total	2900\$

Human Resources Costs:

Table 2.3 Human Resources Costs.

1	150 day x 60\$	9000\$ x 2
	Total	18000\$

Chapter Three

Requirement Analysis

3.1 Functional Requirements

3.1.1 Requirement Definition

The program should be able to:

1. Implement a set of special algorithms for signal processing.
2. Connect Matlab with the web.
3. Perform the processing operation on files determined by the user.
4. Send the result back to the user.
5. Route the result to anywhere in the world.
6. Deal with the expected user errors.
7. Perform reasonable level of security.
8. Control the system remotely.
9. Design and implement friendly graphical user interface.

3.1.1.1. Implement a set of special algorithm for signal processing

The system has a number of special purposes processing algorithms, these algorithms provide a set of operations to process digital signals especially one and two dimension signals such as images and sounds.

3.1.1.2. Connect Matlab with the web

The system should connect the Matlab with the Internet by configuring the Matlab Web Server to accept the demands from the web then submit these demands to

the Matlab. To increase the system management, the system collects the algorithms folders on the server in a manageable way to perform the connection process correctly.

3.1.1.3. User defined file processing

The system should provide the ability to the user to determine the file which contains the signal he wants to process by determining the location of the image or sound file on the client computer to perform the processing algorithms on the specified file.

After the user determines which file will be processed, the system uploads this file to the server then applies the processing operation on this file .

3.1.1.4. Send the result back to the user

After the system processes the file , the result can be presented to the user in two forms, the first form is the result without any compression and the second is the result with compression to reduce the download time of the result, the user can choose which type of output he wants and download it.

3.1.1.5. Rout the result to any where in the world

The processing operation result should not only be accessible by the user but also the user can rout or send this result to any where in the world by sending it as email to any person the user determines. This gives the other person the benefit of the processed file.

3.1.1.6. Deal with the expected user errors

To prevent the errors, the system will have error detection mechanisms to avoid the expected user errors.

These mechanisms prevent large number of problems that can suspend the system and this ensures that the user take his result in a safe way.

3.1.1.7. Perform reasonable level of security

A set of security operation have been taken to secure the system by controlling the user's accounts and preventing unauthorized users to login the system.

The user can create new account and manage this account by configuring his personal information. The system creates a log file for each user to record all user actions.

In addition to that, the system must provide limitation on the algorithm access by putting a price for each algorithm and recording the access action for each algorithm on the database; this gives the system administrator the ability to know the status information for all algorithms.

A bank simulation exists to control credit card accounts that need to cover the cost of the usage of algorithms.

3.1.1.8. Control the system remotely

One of the most advantages and good points of the project is the ability to control the system remotely and from any where in the world.

By this point the administrator can add algorithms easily by configuring the required files in his computer then login to the system via the Internet and upload these files to the specific folder on the server.

The administrator can also access the site and then control the user's account and implement many control functions such as search on user or algorithm and clear the history and many control procedures.

3.1.1.9. Design and implement friendly graphical user interface

The system must have a user friendly interface that provides a useful platform to the user to do all the processes he needs from the site; in addition to this enable the

user to control his account. This site must be easy for the user to send and receive the information to and from the site.

The system also must have a help part to help the users to know which system works.

3.1.2 Requirement Specification

3.1.2.1 Implement a set of special algorithm for signals processing

Software project / Implement a set of special algorithm for signals processing / 3.1.2.1

Function: Implement signal processing algorithms.

Description: Implement algorithms to perform a set of processing operations on signals such as images or sound according to the algorithm job.

Inputs: File contained information to be processed such as image.

Source: Directories in server computer.

Outputs: Processed file as a result of the algorithm.

Destination: The chosen directory where server wants to store the processed file.

Requires: Correct file format; for the image algorithm must use image file the algorithm support and so for sound file .

Pre-condition: Test the file type before processing.

Post – condition: Create new file contain the result without affecting the source.

Side effects: Some times error occurs due to fail in the algorithm job due to error in the source file .

Definition: Software project/ Implement a set of special algorithm for signals processing /3.1.2.1

3.1.2.2. Connect Matlab with the web

Software project / Connect Matlab with the web / 3.1.2.2

Function: Connect Matlab with the web.

Description: Configure the Matlab to accept the demand form the web to apply special algorithm on the specified file to process the data.

Inputs: Matlab web server configuration files.

Source: Web server computer.

Outputs: Result of the algorithm connected with the web pages.

Destination: Result of the algorithm output to specified web page.

Requires: Correct configuration of the Matlab web server files such as webconfig file, and save the correct structure of these files.

Pre-condition: Test the existence of the algorithms listed in the configure file.

Post – condition: The output directed to the directory specified the configure file.

Side effects: Any error of configuration will cause the system processing failed.

Definition: Software project / Connect Matlab with the web / 3.1.2.2

3.1.2.3. Perform the processing operation on file determined by the user

Software project / Perform the processing operation on file determined the user /

3.1.2.3

Function: User control the source file location which will be processed.

Description: The user control the source files location within the client computer, this file will be processed by using specific algorithm.

Inputs: Location of the source file.

Source: Client computer.

Outputs: Copy or upload the file from the client computer to the server computer.

Destination: The uploaded file stored in the user directory in the server.

Requires: Correct determination of the source, by inter valid file destination and name.

Pre-condition: Test the existence source.

Post – condition: The output directed to the location determined by the server in the user folder in the server.

Side effects: Any error will occur if user fails while the direction process of the source and destination directory or upload error.

Definition: Software project / Perform the processing operation on file determined the user / 3.1.2.3

3.1.2.4 Send the result back to the user

Software project / Send the result back to the user / 3.1.2.4.

Function: Send the result back to the user.

Description: After the system processes the file the result presented to the user in two forms, the first form is the result without any compression and the second is the result with compression to reduce the download time of the result.

Inputs: Files output from the Matlab.

Source: The processing algorithm the user selected.

Outputs: The result represented in compressed and uncompressed form.

Destination: The user selects the destination of the downloaded file on his computer.

Requires: Determine the result version the user needs to download.

Pre-condition: None.

Post – condition: The file uploaded to the user computer.

Side effects: Take time to download the result.

Definition: Software project / Send the result back to the user / 3.1.2.4.

3.1.2.5. Rout the result to any where in the world

Software project / Rout the result to any where in the world / 3.1.2.5.

Function: Rout the result to any where in the world.

Description: The processing operation result not only accessible by the user but also the user can rout or send this result to any where in the world by sending it as email to any person he determines.

Inputs: Insert email name.

Source: Web page that have two field for type two emails.

Outputs: Email with attachment.

Destination: The user selected email.

Requires: Check the check box beside the email text field.

Pre-condition: Type valid email.

Post – condition: Email transmitted to the specified email names.

Side effects: Send emails to invalid destination if the user type error emails names.

Definition: Software project / Rout the result to any where in the world / 3.1.2.5.

3.1.2.6. Perform reasonable level of security

Software project / Deals with the expected user errors/ 3.1.2.6.

Function: Create errors detection mechanisms.

Description: Built errors detection mechanisms to avoid the expected user errors.

Inputs: Errors from the users.

Source: User input

Outputs: Error messages to the user.

Destination: Control the user's actions; by determining the validity of these actions.

Requires: Error detection codes in any where the user may perform an error.

Pre-condition: None.

Post – condition: None.

Side effects: None.

Definition: Software project / Deals with the expected user errors/ 3.1.2.6.

3.1.2.7. Perform reasonable level of security

Software project / Perform reasonable level of security / 3.1.2.7.

Function: Built authentic use of the site.

Description: Built security system to control all the action done in the site by control the users account and the algorithms access.

Inputs: Account number and password.

Source: User input and the system database.

Outputs: Give permissions to the users.

Destination: Determine the authenticated users and the accessibility on the algorithms.

Requires: Complete user information login form.

Pre-condition: None.

Post – condition: None.

Side effects: Take time from the user to complete the login form and registration form

Definition: Software project / Perform reasonable level of security /3.1.2.7.

3.1.2.8. Control the system remotely

Software project / Control the system remotely / 3.1.2.8.

Function: Control the system remotely.

Description: The system provides a set of tools for the administrator to control the system remotely, by search tools about users and algorithms, and the ability to add algorithms remotely to the system.

Inputs: A set of action according to his needs.

Source: Client computer any where in the world, with user have administration permission.

Outputs: Result form the system according to the administrator needs.

Destination: Perform the administrator demands.

Requires: Administration permissions.

Pre-condition: None.

Post – condition: None.

Side effects: This part gives the administrator some of administrator's tools not all tools.

Definition: Software project / Control the system remotely / 3.1.2.8.

3.1.2.9. Design and implement user friendly web application

Software project / Design and implement graphical user interface/ 3.1.2.9.

Function: Implement user friendly web application.

Description: Implement easy powerful and user friendly interface.

Inputs: The website visual elements.

Source: User input and the system database.

Outputs: Web application interface.

Destination: None

Requires: Visual element.

Pre-condition: None.

Post – condition: None.

Side effects: None.

**Definition: Software project / Design and implement graphical user interface/
3.1.2.9.**

3.2 Non-Functional Requirements

The non-functional requirements are responsible for the requirements that is not concerned with the need of the software or any thing related to the functionality of the software program like the requirement concerned with time, team properties and so on, a classification done for the non-functional requirements into three classifications:

3.2.1 Product Requirements

- a. Easy to access and login.
- b. High speed system response.
- c. Easy to use.
- d. Clear and good management of the algorithm in the site.

3.2.2 Process Requirement

The software should be available at the end of the winter semester of the year (2003-2004).

3.2.3 External Requirements

- a. Each site owner can use his own algorithms and determine his algorithm price and number.
- b. The system should be able to deal with other kind of processing (the system can improve to work with other processing fields not only signal processing of course by changing some system element such as algorithms).

Chapter Four

System Design

4.1 Design Models

4.1.1 Introduction

Design models are the representative models of the software system that describe the system in the mode of figures and designs which give the reader the knowledge of the system at a glance and proceed without involving through the system details or testing. These models include context model, data flow diagram, structural chart, data and control analysis chart and the flow chart of the system.

4.1.2 Context (Relation with Other Systems)

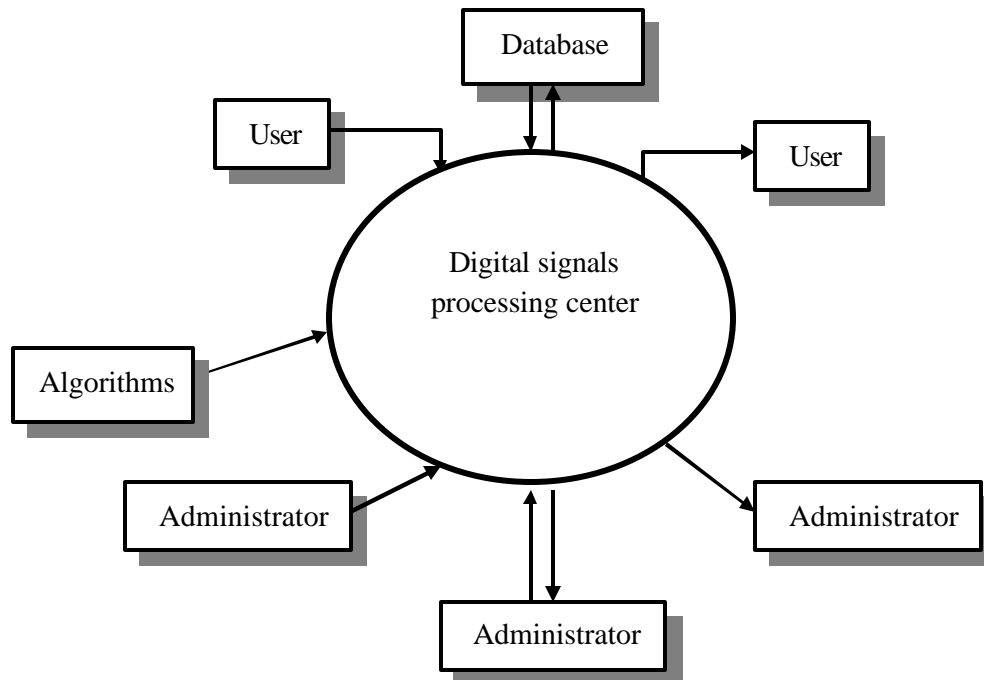


Figure 4.1 Context Diagram.

4.1.3 System Flow Diagram

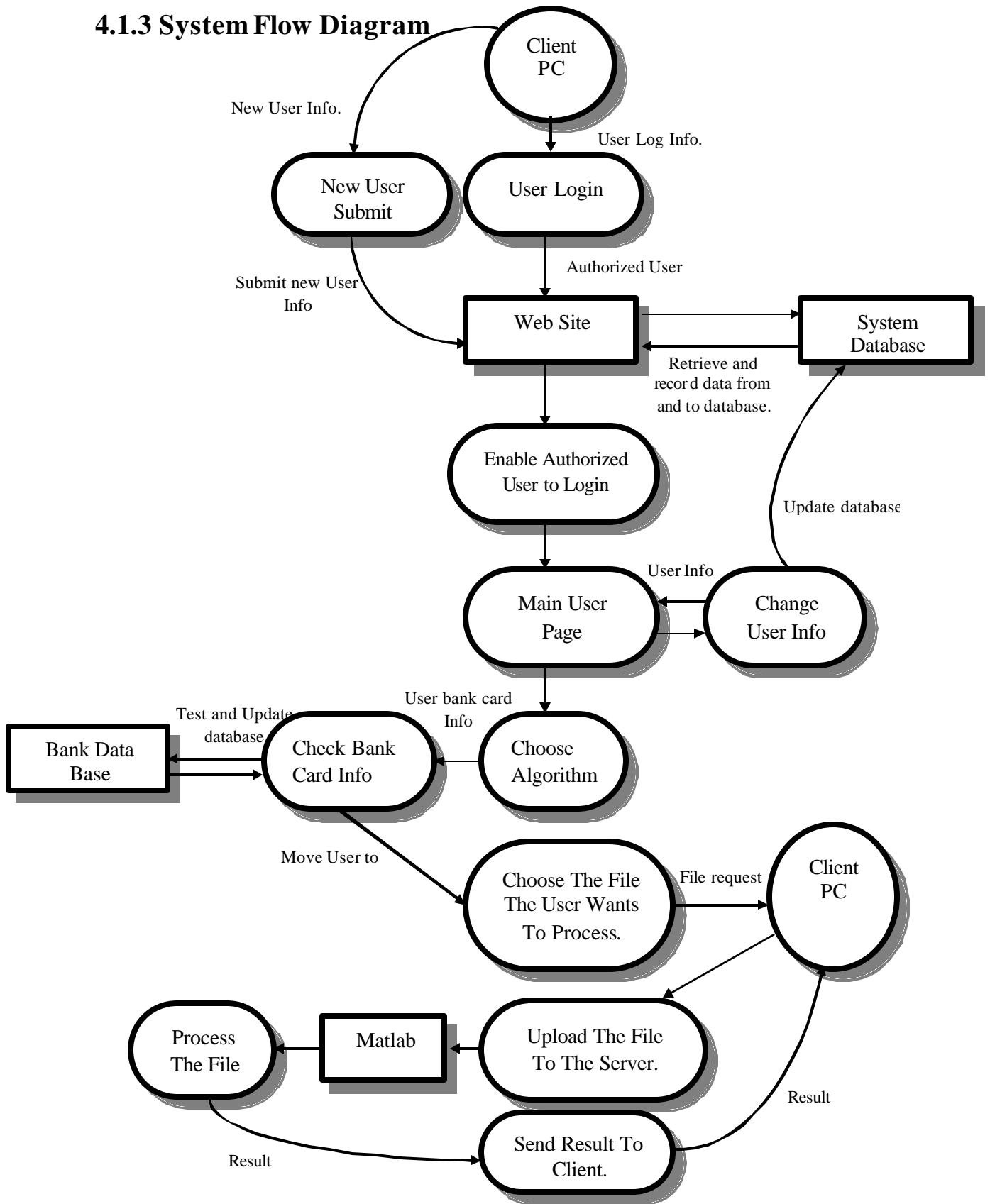


Figure 4.2 System Flow Diagram.

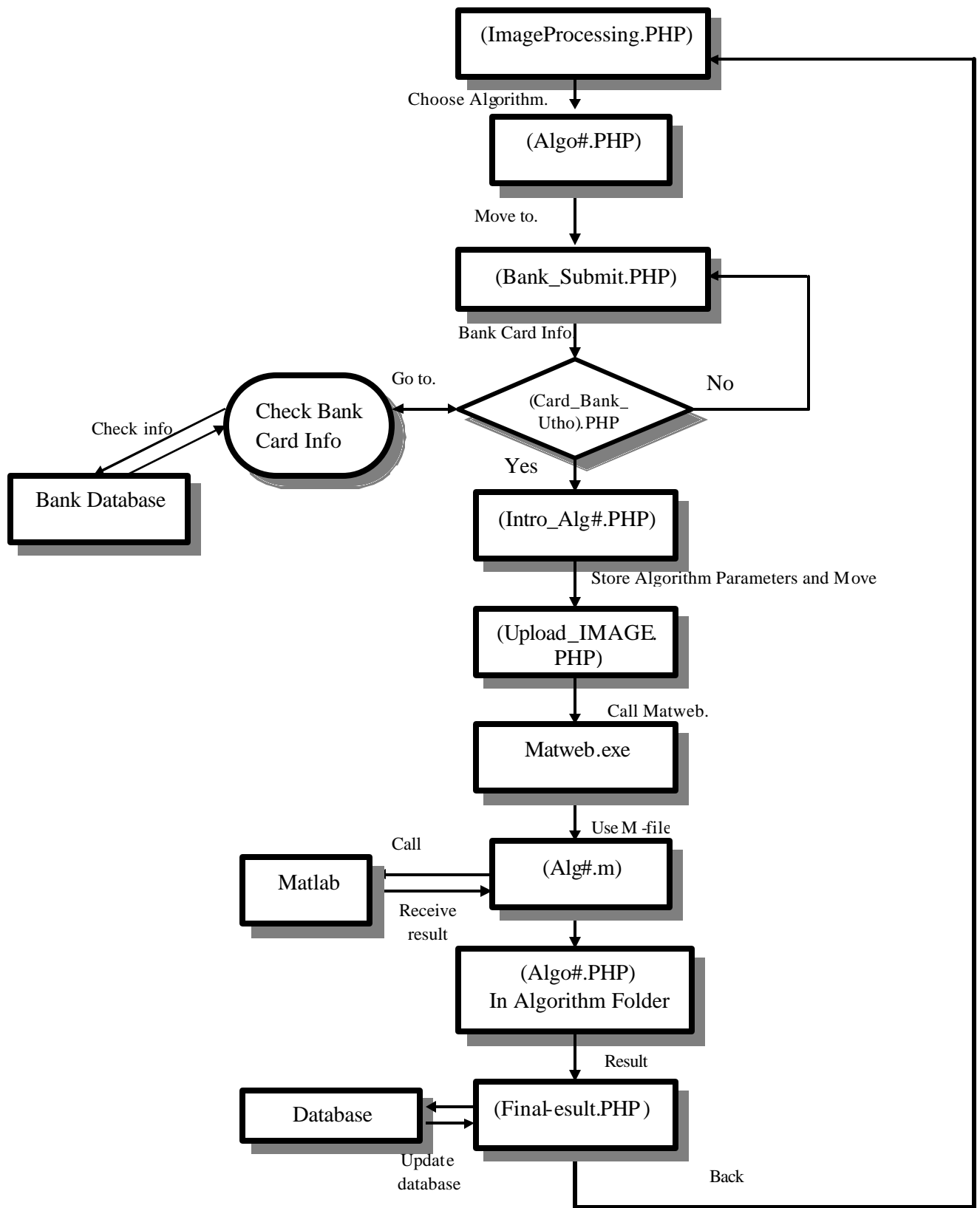


Figure 4.3 Image Processing Algorithms Flow.

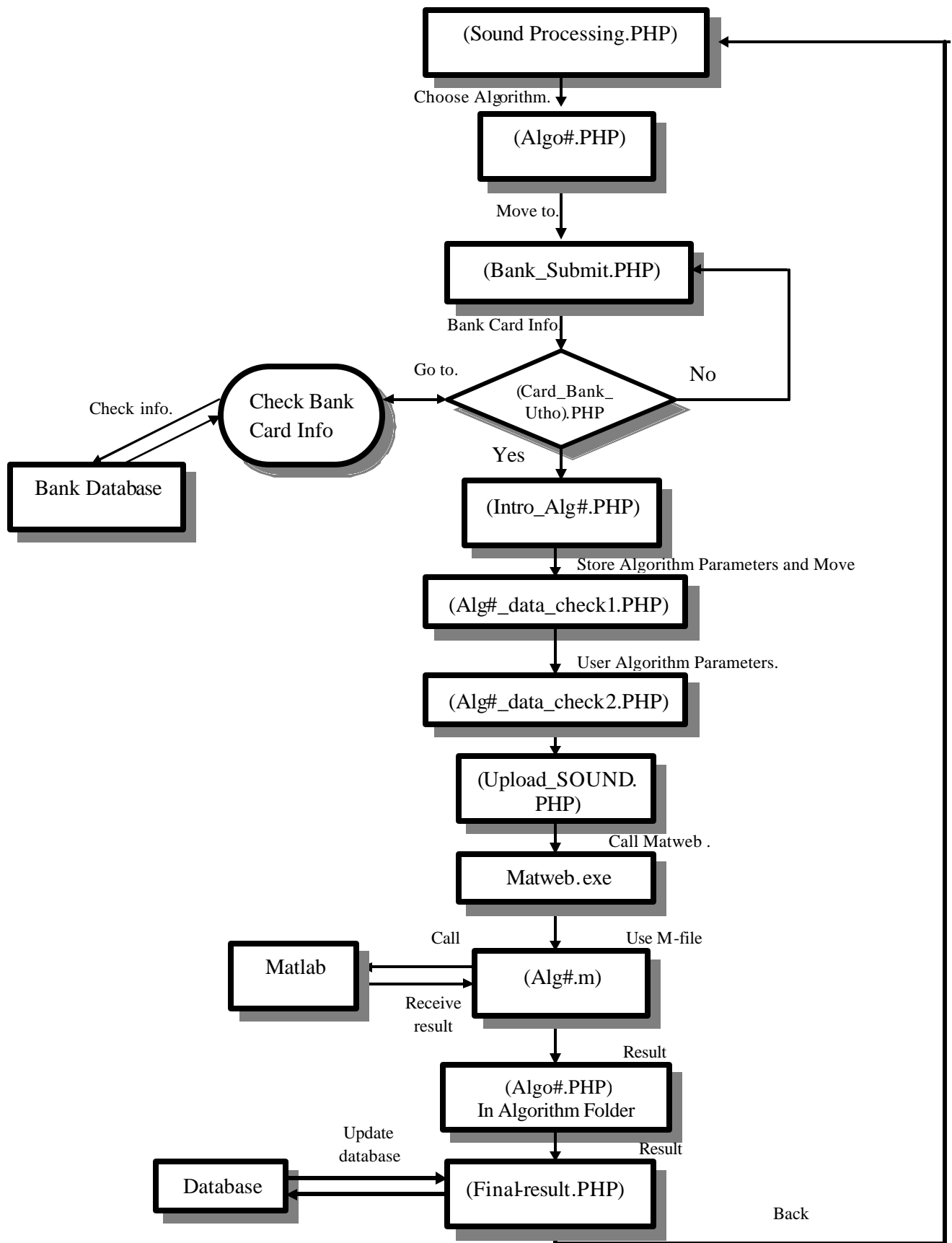


Figure 4.4 Sound Processing Algorithms Flow.

4.1.4 Structure Chart

This diagram shows more specification information about the main component of the system which the context diagram represented.

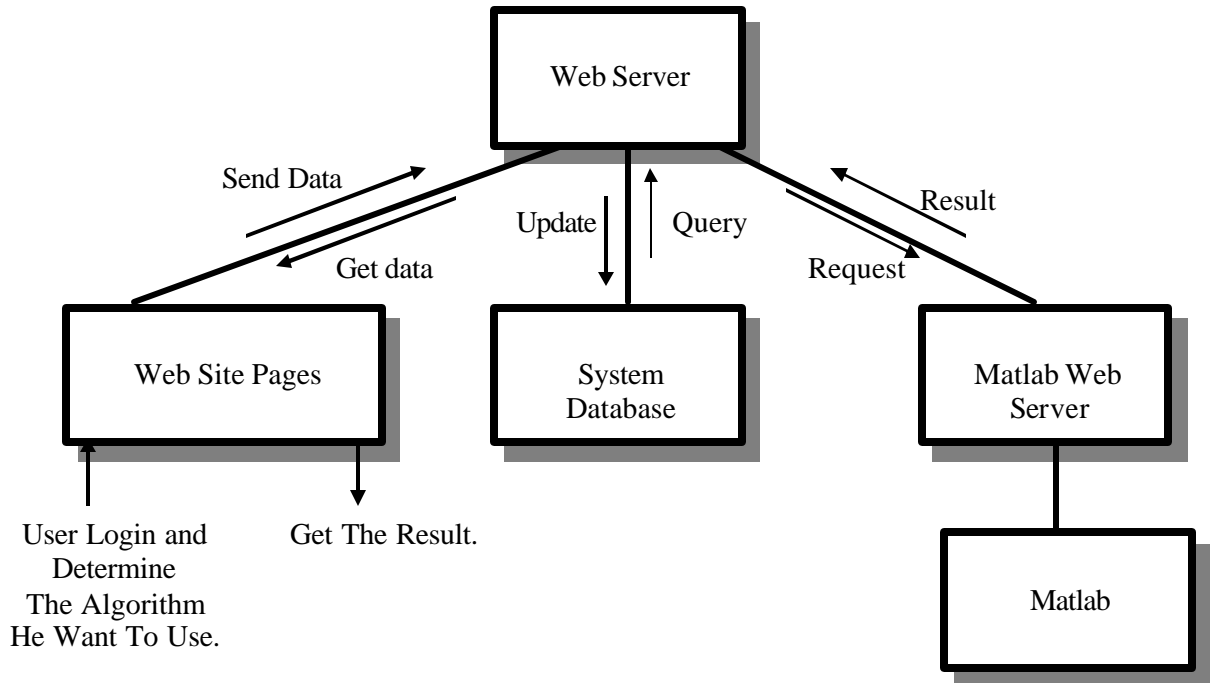


Figure 4.5 Structure Chart.

4.1.5 Data and Control Analysis

The figures below show the data and control analysis for the system client and the remote system administrator.

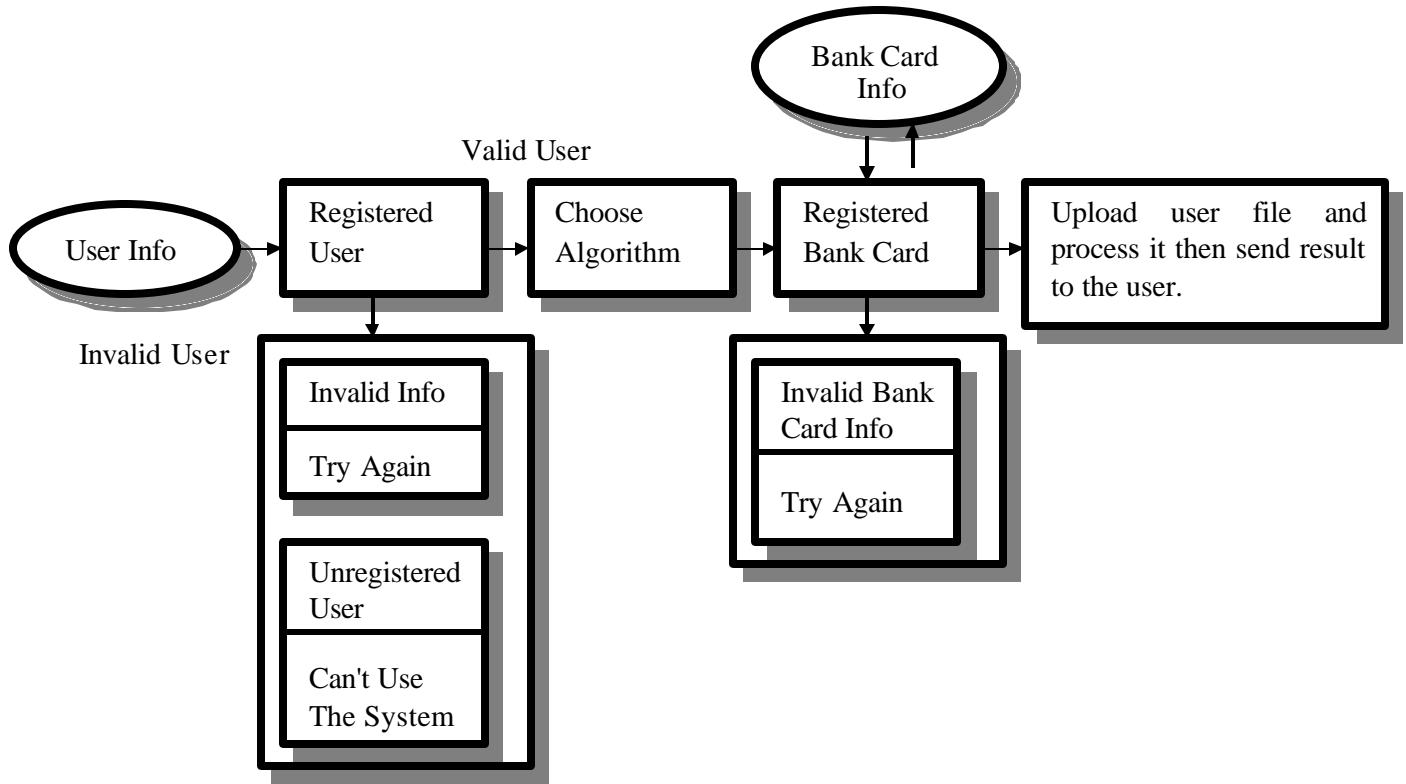


Figure 4.6 Data and Control Analysis for Client.

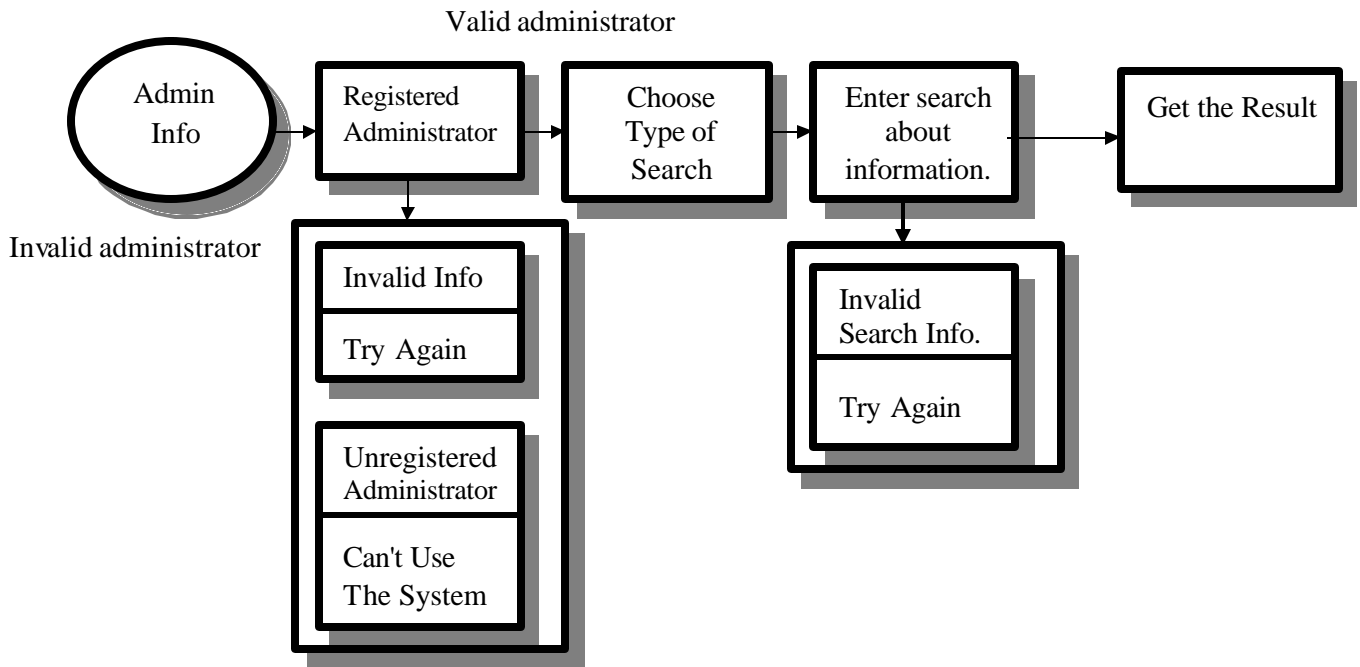


Figure 4.7 Data and Control Analysis for administrator (Searching).

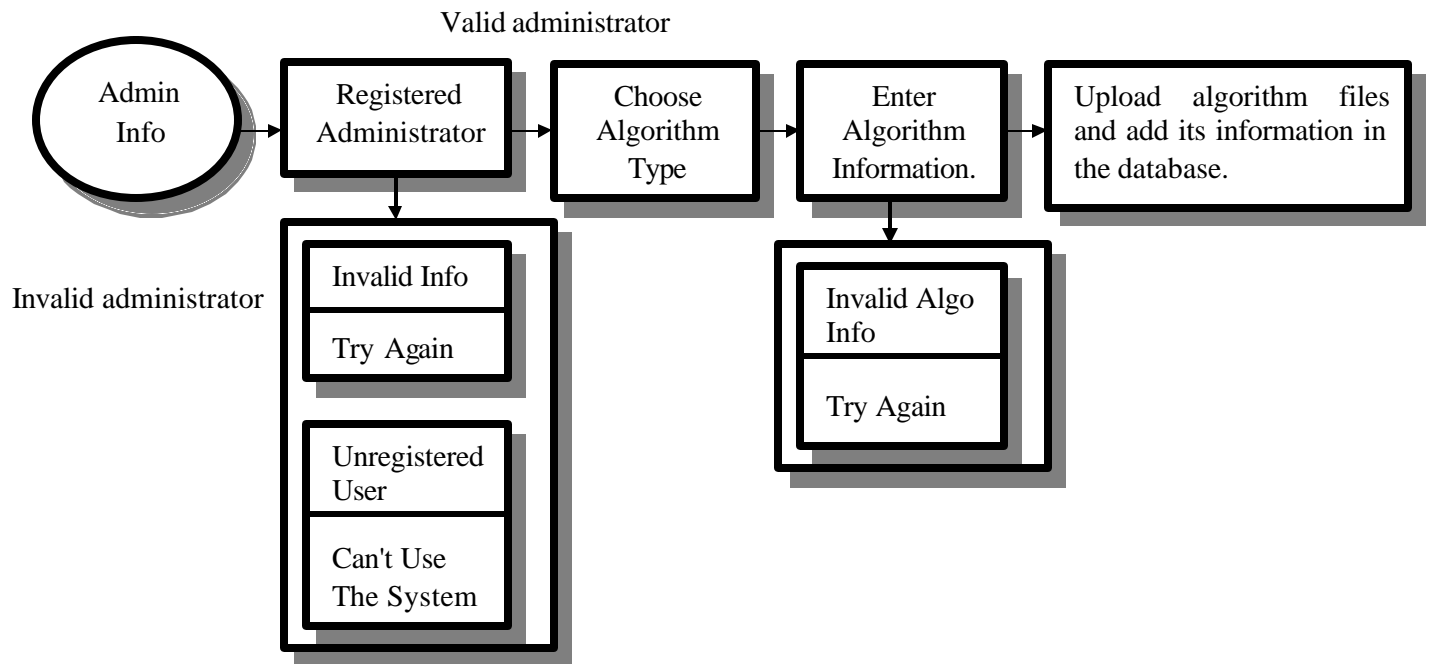


Figure 4.8 Data and Control Analysis for administrator (Add Algorithms).

4.2 Project Databases Design

4.2.1 Introduction

When we talk about large and strong web applications, we must talk about the database, in spite of the fact that PHP language support all database engines but the interest with MySQL database because it is free and available in most database server.

Also the ability to efficiently store and retrieve large amounts of information done by storing this information in database, and this resulting in faster and more flexible of storing and retrieval of data, adding, searching and sorting.

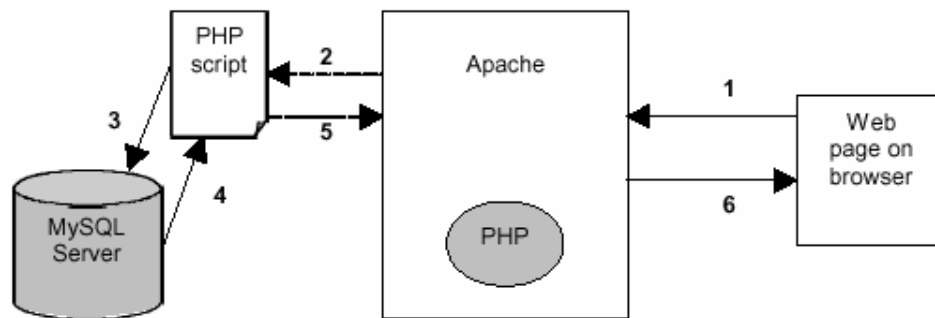


Figure 4.9 PHP, Apache and MySQL Connection.

MySQL, the most popular Open Source SQL database, is developed, distributed and supported by MySQL AB. MySQL AB is a commercial company, founded by the MySQL developers, that builds its business providing services around the MySQL database.

4.2.2 MySql

In this section a set of information about MySql will be discussed to increase the understanding about this subject.

4.2.2.1 MySql is a Database Server

The MySql Database Server is very fast, reliable and easy to use. MySql Server also has a practical set of features developed in close cooperation with MySql users.

4.2.2.2 MySql is a Database Management System

A database is a structured collection of data. It may be anything from a simple shopping list to vast amounts of information in a network. To add, access and process data stored in a computer database you need a database management system such as MySql Server.

4.2.2.3 MySql is a Relational Database Management System

A relational database stores data in separate tables rather than putting all the data in one big storeroom, this adds speed and flexibility.

4.2.2.4 MySql is Open Source Software

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySql software from the Internet and use it without paying anything.

4.2.3 System Database Specification

The system has four database tables these are:

- ? Algorithm Table.
- ? User Table.
- ? Administrator Table.
- ? Bank simulation Table.

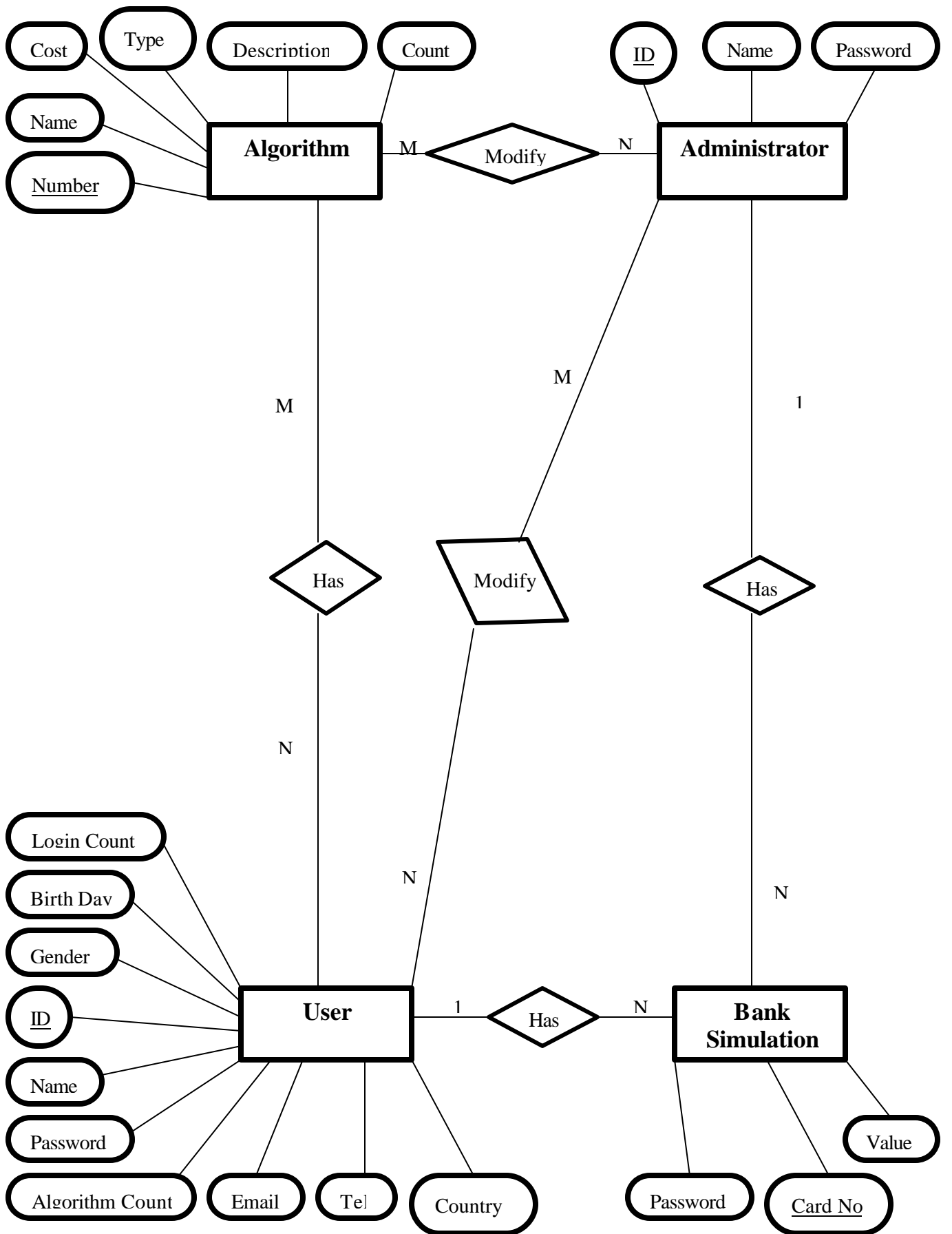


Figure 4.10 ER-Model

4.2.3.1 Algorithm Table:

This database table contains information about every algorithm in our web application; the number of the algorithms depends on the algorithm the owner of the system want to present to the user initially we put 12 algorithm in image and sound processing. The fields in this table are:

ALG_NUM: This field contains the number of algorithm; this number is sequentially started from 1 and incremented by one every time the administrator adds new algorithm.

ALG_NAME: This field contains the name of algorithm; this name starts with ALG concatenate with the number of algorithm (ALG1, ALG2, etc).

ALG_COUNT: This field contains the count of iteration this algorithm was used; this number is sequentially started from 0 and incremented by one every time the algorithm was used.

ALG_DES: This field contains the description of the algorithm and shows the name and method of algorithms.

ALG_TYPE: This field contains the type of the algorithm (image, sound).

ALG_COST: This field contains the cost of the algorithm; this cost is taken off from the user's credit card when he uses the algorithm.

4.2.3.2 User Table:

This table contains information about every user in our web application, the fields in this table are:

USER_ID: This field contains the ID of the user; this number is sequentially started from 990000 and incremented by one every time new user registers in our web application.

USER_NAME: This field contains the name of the user; this field accepts only characters value.

USER_PASSWORD: This field contains the password of the user's account.

USER_ALG_COUNT: This field contains the count of iteration that the user use an algorithm, this number is sequentially started from 0 and incremented by one every time the user uses an algorithm.

USER_COUNTRY: This field contains the name of the user's country; this field accepts only characters value.

USER_TELEPHONE: This field contains the telephone of the user; this field accepts only numerical value.

USER_E_MAIL: This field contains the E-mail of the user; this field accepts only correct E-mail (UserName@ServerName.com, UserName@ServerName.org, UserName@ServerName.edu)

USER_GENDER: This field contains the gender of the user Male or Female.

USER_BIRTH_DAY: This field contains the day of the user's birthday.

USER_BIRTH_MONTH: This field contains the month of the user's birthday.

USER_BIRTH_YEAR: This field contains the year of the user's birthday.

LOGIN_COUNT: This field contains the count of iteration that this user login in our web application, this number is sequentially started from 0 and incremented by one every times the user bgin.

4.2.3.3 Administrator Table:

This table contains information about the administrator in our web application, the administrator has large permissions, and the fields in this table are:

ADM_ID: This field contains the ID of the administrator; the administrator must login in the database and register manually to ensure the security.

ADM_NAME: This field contains the name of the administrator.

ADM_PASSWORD: This field contains the password of the administrator's account.

4.2.3.4 Bank Simulation Table:

This database represents bank simulation because the user cannot use the system without paying the cost of the algorithm usage. Which contains information about the credit card in the bank, the fields in this table are:

Card_No: This field contains the number of the credit card.

Value : This field contains the value of the credit card.

Password: This field contains the password of the credit card.

Chapter Five

Implementation

The implementation of the project involves a number of steps or phases some of which are performed serially others in parallel; this section describes the implementation steps of project.

5.1 General View:

Chapter four shows many figures and diagrams to increase the knowledge about the system, but in this chapter more detailed implementation steps are represented.

This chapter discusses each part separately and connects it to the other parts, more programming code about some subject in this chapter will be found in the appendices part in this documentation.

5.2 Prepare The Platform of The System:

Before operating the system a number of configuration steps must be taken to prepare the suitable platform to operate the system.

5.2.1 Install And Configure The Web Server:

We want to operate computer as a server to enable the Internet users to access the web application side of the system. So the apache web server was installed on the computer, to get the more effective performance it's preferred to use the last version of this software, in this project apache 1.2.32 is chosen to work with.

This server gives the system needs. This server works in good way with Matlab web server and many tests were applied on this server to ensure that it is

compatible with the Matlab software; in addition to that the apache server can work with the UNIX system and so the Matlab and PHP do the work on this server may enable to operate the system we built on UNIX platform not only on windows of course with small modification.

Note that the directory of apache server on the computer hard drive is "c:\apache".

There are two versions of this software one of them works with windows platform and the other works with the UNIX platform, because the system works in windows platform we use the windows compatible version of this software.

5.2.2 Install PHP Compiler:

It's necessary to install PHP web programming language compiler to compile and run the PHP codes.

5.2.3 Install The Database Software:

In the system the MySQL database was selected to work with, so this database system must be installed. The system uses this type of database system because it works effectively with the apache web server and we see it gives the ability to perform the entire database operations needed in this project in efficient way.

There are two versions of this database management system one of them works with windows platform the works work with the UNIX platform, because the system works in windows platform we use the windows compatible version of this software.

5.2.4 Install and Configure The Matlab:

The version of the Matlab had been chosen in this project is Matlab 6.5 that have new feature called Matlab web server which provides the ability to connect the

Matlab with the web and the following steps show configuration steps of the Matlab and Matlab web server to work in the project:

The first step had been done is the installation of Matlab and insure that the Matlab web server option is selected to install it to enable connect the Matlab with the Internet. It's preferred to install all Matlab libraries to make it able to perform many of processing operation it can perform or install at least the parts of Matlab that the system algorithms needs.

To configure the Matlab web server follow the following steps:

- Copy matweb.exe and matlabserver.exe found in "<Matlab>/webserver/bin/ win32 ", to the cgi-bin folder in the web server directory "c:\apache\ ".
- Copy Matweb.conf in <Matlab>/toolbox/webserver/wsdemos to the cgi-bin folder in the web server directory "c:\apache\ ".
- Configure the Matweb.conf file by add the names of the algorithm and the algorithm folder as shown in the appendix A.

5.3 Web Site Design and Implementation:

After preparing and configuring all elements necessary to the systems the work group started working with the implementation of the web site.

The system works with the active server pages to create the web site to make the site more dynamic and powerful.

Through designing and implementing time of the system the Macromedia Dreamweaver software is selected to build the site which enables us to built friendly user interface.

5.3.1 Site Block Diagram:

To clear and increase the understanding of the action that the user can do in the site the following general block diagram shows allowed action the user can do.

All pages belong to the site stored in (C:\apache\htdocs\project); also in this directory a set of folders belong to the system such as (Algorithm folder) which has the Matlab M-files and web pages for each algorithm the Matlab needs.

The project directory has a folder for each algorithm and these folders are named as the algorithm's name, the discussion about these folders will come in the next topic subjects section.

The folder (SW PROJECT DATABASE) in the MySql folder within the server directory has the system database including all tables.

The (USER_LOGIN) folder store the log file for the users of the system and his temporary files, each user has a log file named as the user ID.

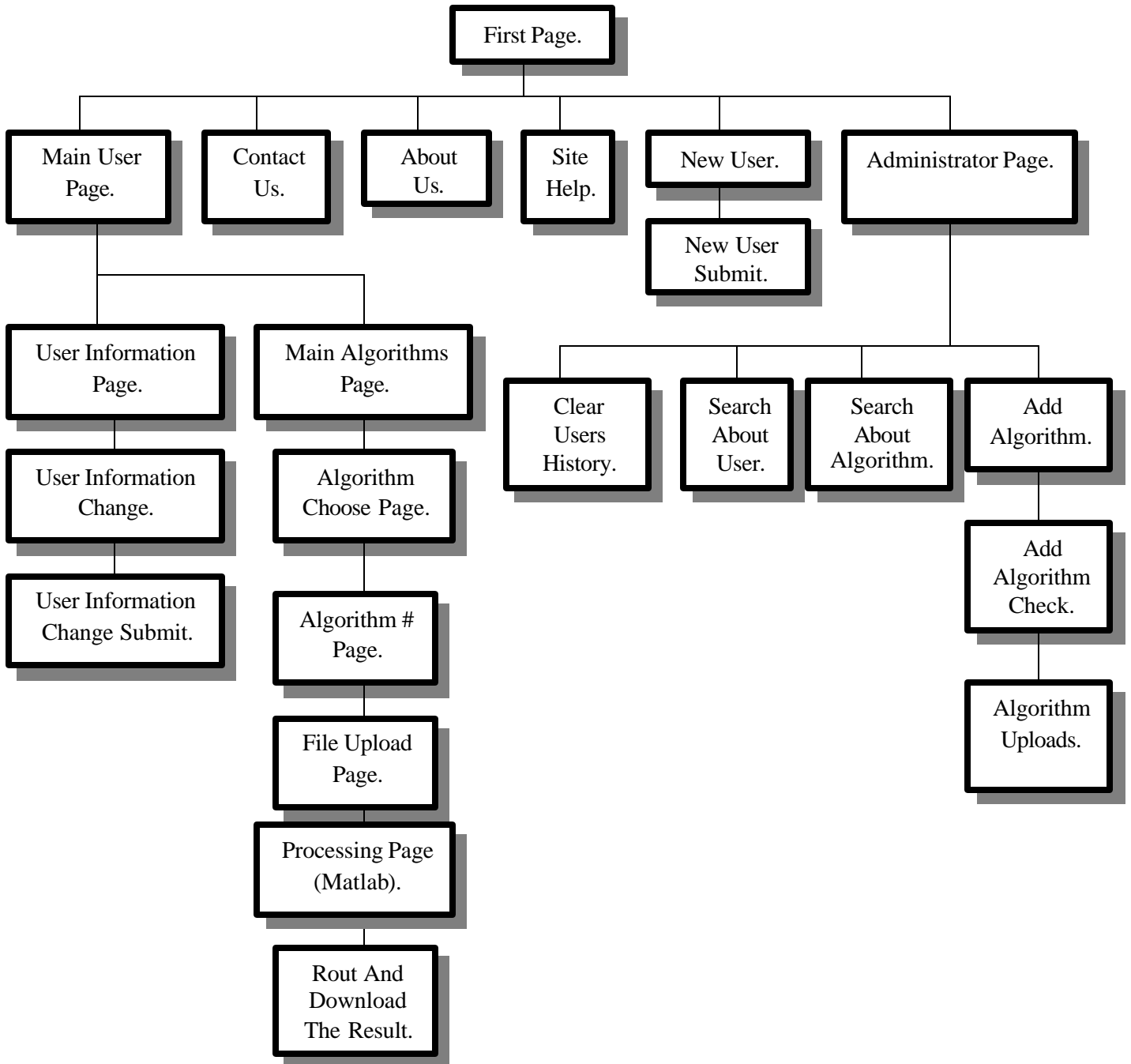


Figure 5.1 The Site Map.

5.3.2 General Site Pages Specification:

After viewing the general block diagram we now want to explain each page and its content.

The codes of these pages will be found in the university library but sample of the code listed in the appendices part of the documentation.

5.3.2.1 Index.php "The index page of the site":

This page is the first page the user sees if he visits the site; it has a link to (Page1.php) which the user can interact with the system.

5.3.2.2 Page1.php "The main page of the site":

This is the main page the user sees when he starts using the site; this page contains a set of element which enable the user to perform some action.

The user can:

- Access (Site Help) page, (Contact Us) page, and (About Us) page.
- Access (New User) page if the user wants to register as new user.
- Access (User Login) page if the user wants to login to the site. After filling the user ID and password filed.

5.3.2.3 Site_Help.PHP "User help page":

This page shows the help about the system and much information to make the users work with the system in good way and make them able to use all system functions and operations.

5.3.2.4 About_Us.PHP "Work group information page":

This page is used to show complete information about the work group, and other necessary information about the system creators.

5.3.2.5 Contact_Us.PHP "Contact with work group page":

This page enables the user of the site to contact with the work group by email or other communication media.

5.3.2.6 New_User.PHP "Register as new user page":

By login to this page the user can register as new user to enable him to access the system and used the site properties.

The user must enter his personal information and in valid way. The information entered will pass to other PHP file that test the validity of the data entered this file is named (New_User_Submit.php).

5.3.2.7 New_User_Submit.php "Tests the validity of data":

This page tests the entered information by (New_User.PHP) page and ensures that the data is correct, any error in this data will stop the submitting process and display error message to the user.

When the data accepted by the system the user information will be added to the database and display page shows the user ID and other necessary information to enable him to login to the system.

The code of this page doesn't allow the user to update the page to prevent duplication of the data.

5.3.2.8 User_Login_Check.PHP "User information check":

From (Page1.php) the user can login to the system by type his ID and password then press login key, now the system will call the (User_Login_Check.PHP) page to check the authentication of this user by test the ID and the password from the User_Table in the database.

If the user is authorized the system updates the database and login file of the user by call (UPDATE_ACCESS.php) page; then it will move the user to the (User_Main_Page.php).

5.3.2.9 Update_Access.PHP "Site action information":

This page will update the log file and database record of the user according to the action performed by the user such as login or using some algorithm.

The update of the database include update the login counter of the user and the counter of the algorithms use.

The update of the log file includes recording the user action such as the login time, date and the algorithm number.

5.3.2.10 User_Main_Page.PHP "The main page of project":

This is the main user page that the user can see his information by submitting to (User_Info_Page.PHP) page, all action the user has done in this system will be displayed to him as a table in this page.

The user can choose the type of processing he wants by clicking on the processing type link in the page; the user can choose:

- Image Processing Algorithms by visiting (ImageProcessing.PHP) page.
- Sound Processing Algorithms by visiting (SoundProcessing.PHP) page.

5.3.2.11 User_Info_Page.PHP "To display the current user information":

This page is connected with the database and gets the current user information and lists it as table. There is a key in this page which allows the user to change his information by visiting (User_Info_Change.PHP) page.

5.3.2.12 User_Info_Change.PHP "To enter new user information":

This page allows the user to enter the renewed information about him and submit this information.

This information is submitted to (User_Info_Change_Submit.PHP) page to test the validity of the data.

5.3.2.13 User_Info_Change_Submit.PHP "Information validity checks":

This page test is the information entered by the user to check the validity of this data and then allow the user to re login to the site with the new information after updating the database information.

Any error in the user information message will display the type of the error to the user. If the user enters a valid data, the system accepts this data and updates the database record of the user.

5.3.2.14 ImageProcessing.PHP "Image processing algorithms page":

This page scans the algorithm table in the database and lists the name of the image processing algorithms the system can work with. And allows the user to choose which algorithm he wants; the page not only lists algorithms name but also lists its cost and brief information.

5.3.2.15 SoundProcessing.PHP "Sound processing algorithms page":

This page scans the algorithm table in the database and lists the name of the sound processing algorithms the system can work with and allows the user to choose which algorithm he wants; the page not only lists algorithms name but also lists its cost and brief information.

5.3.2.16 My_Upload_IMAGE.PHP "Image file upload":

This PHP file performs the image file upload process to copy user defined file from the client side to the server side and stores this file in the user folder in the server; the storing of this file is done by name depending on the algorithm name. The important parts of this page code are found in the appendix B.

5.3.2.17 My_Upload_SOUND.PHP "Sound file upload":

This PHP file performs the sound file upload process to copy user defined file from the client side to the server side and stores this file in the user folder in the server; the storing of this file is done by name depends on the algorithm name. The important parts of this page code can be found in the appendix C.

5.3.2.18 Bank_Submit.PHP "Bank card information form":

This page allows the user to enter the card number and password for his credit card, this page called by the algorithms pages to reduce the cost of the algorithm from the card value.

5.3.2.19 Card_Bank_Utho.PHP "Bank card Check":

This PHP file is used to test the validity of the user bank card and see if he has enough money to use the algorithm.

5.3.2.20 Bank_Error.PHP "Bank card error message":

This page displays erroneous messages to the user, if there is a problem in the bank card.

5.3.2.21 Final_Result.PHP "Link to the result page":

This page has link to the result to download it by the user. The result is presented in two forms compressed and uncompressed. And also this page call page (Send_Result_To.PHP) to send the result to some email.

5.3.2.22 Send_Result_To.PHP “Sends the result by email”:

This PHP file is used to send emails containing the result as attachment. These messages will be sent by the mail server. The code of this file can be found in appendix D.

Note: All these pages found in the directory (C:\apache\htdocs\project).

5.3.3 Algorithm Pages Specification:

This section displays the method which the system performs to deal with the algorithms pages.

There are two groups of algorithm files; the first is images processing and the other is sound processing.

5.3.3.1 Image Processing Part:

All algorithms of this type have the same construction, the image processing part needs the following files:

- Algo (Num). PHP.
- Intro_Algo (Num). PHP.
- Alg (Num).m (In algorithm folder).
- Algo (Num).PHP (In algorithm folder).

The discussion of this part will be on the algorithm number (One) which is image type algorithm. All other image algorithms are built using the same method.

5.3.3.1.1 Algo1. PHP "Algorithm1 information ":

This page displays the information about algorithm1 including pictures to increase the understanding of the work of this algorithm. When the user wants to access the algorithm, the system will transfer the user to the (Bank_Submit.PHP) page to check the credit bank card.

Note: This page can be found in the directory (C:\apache\htdocs\project\Alg1).

5.3.3.1.2 Intro_Algo1. PHP "Complete algorithm1 information":

This page allows the user to determine the location of the file on the client computer to be uploaded to the server to process it. Any error in the file to be uploaded the process will not be complete until corrected the error, then after uploading the file the system will ask the user if he wants to continue to process the file. This is done by calls the algorithm1 Matlab M-file in the algorithm folder this file called (Alg1.m) and process the file by Matlab web server.

Note: This page can be found in the directory (C:\apache\htdocs\project\Alg1).

5.3.3.1.3 Alg1.m "Algorithm1 M-file":

Each algorithm has M-file found in folder (Algorithms), in this file is a set of elements. These elements are divided to three parts:

- A- Input data from the caller web page.
- B- Processing part.
- C- Output data to the web page.

The M-file not only processes the data but also performs compression operation on the result to reduce the result download time. The code of this M-file is in appendix E.

Note: This page can be found in the directory (C:\apache\htdocs\project\Algorithms).

5.3.3.1.4 Algo1.PHP “The web page that receive M -file output”:

This page receives the output from the processing algorithm M-file and sends this result to the user by sending the result to the (Final_Result.PHP). The cost of these algorithms will be taken from the user if he clicks on the (Get Result) key.

Note: This page can be found in the directory (C:\apache\htdocs\project\Algorithms).

5.3.3.2 Sound Processing Part:

All algorithms of this type have the same construction. The image processing part needs the following files:

- Algo (Num).PHP.
- Intro_Algo (Num).PHP.
- Alg (Num)_data_check1.PHP.
- Alg (Num)_data_check2.PHP.
- Alg (Num).m (In algorithm folder).
- Algo (Num).PHP (In algorithm folder).

The discussion of this part will be on the algorithm number (Seven) which is sound type algorithm. All other image algorithms are built using the same method.

5.3.3.2.1 Algo7. PHP "Algorithm7 information ":

This page displays the information about algorithm7. When the user wants to access the algorithm, the system will transfer the user to the (Alg7_data_check1.PHP) page to complete the user defined parameter the algorithm want to work with.

Note: This page can be found in the directory (C:\apache\htdocs\project\Alg7).

5.3.3.2.2 Alg7_data_check1. PHP “Complete algorithm7 parameter information”:

This page allows the user to determine values of the algorithm parameters, and asks the user to re input the value if there is an error by submitting the data to the (Alg7_data_check1.PHP) page.

Note: This page can be found in the directory (C:\apache\htdocs\project\Alg7).

5.3.3.2.3 Alg7_data_check2. PHP “Check algorithm7 parameter”:

Test the data value from the (Alg7_data_check1.PHP) page and prevent the user to enter unwanted values to the algorithm to prevent process error.

If the data is valid, the system will test the bank card information as shown in the image processing part.

Note: This page can be found in the directory (C:\apache\htdocs\project\Alg7).

5.3.3.2.4 Intro_Alg7. PHP “Complete algorithm7 information”:

This page allows the user to determine the location of the file on the client's computer to be uploaded to the server to process it. If there is any error in the file to be uploaded the process will not be completed until the error is corrected. Then after uploading the file the system will ask the user if he wants to continue to process the file. This is done by calling the algorithm7 Matlab M-file in the algorithm folder this file is called (Alg7.m) and process the file by Matlab web server.

Note: This page can be found in the directory (C:\apache\htdocs\project\Alg7).

5.3.3.2.5 Alg7.m “Algorithm7 M-file”:

Each algorithm has M-file found in folder (Algorithms). In this file there is a set of elements, these elements are divided to three parts:

- D- Input data from the caller web page.
- E- Processing part.
- F- Output data to the web page.

The M-file not only processes the data but it also performs compression operation on the result to reduce the result download time. The code of this M-file in appendix F.

Note: This page will be found in the directory (C:\apache\htdocs\project\Algorithms).

5.3.3.2.6 Algo7.PHP “The web page that receive M -file output”:

This page receives the output from the processing algorithm M-file and sends this result to the user by sending the result to the (Final_Result.PHP). The cost of these algorithms will be taken from the user if he clicks on the (Get Result) key.

Note: This page found in the directory (C:\apache\htdocs\project\Algorithms).

5.3.4 Administrator Pages Specification:

The site provides a set of operations remotely to increase the management of the system, these operations are search about the users and perform some operations on their accounts, the administrator can also search about algorithms and the administrator of the system can add new algorithm to the system remotely from any where by configuring the needed files and then uploading them to the server as we will see in the next topics.

5.3.4.1 Admin_Page.PHP “Administrator page”:

This page can be accessed by typing the administrator’s ID and password in (page1.PHP), this page has many parts each to perform specific job.

The administrator can:

- A- Clear user history: The administrator can clear the history of all users or a specific user.
- B- Search about user: Search about a user or list all users
- C- Search about algorithms: Search about algorithm or list all algorithms.
- D- Add new algorithm to the system.

5.3.4.2 Clear_User_History.PHP “Clear user's history”:

This page clears the user's history information; the administrator can clear all users' history or clear the information to specific user by typing his ID.

This page can be accessed when the user clicks on the (Clear History) key in the (Admin_Page.PHP), and checks the check box (Clear_User_History) to clear all users' history.

5.3.4.3 Admin_User_List .PHP “List user's information”:

This page lists all users' accounts and their information to the administrator. This page accessed when the user clicks on the (All Users) key in the (Admin_Page.PHP).

5.3.4.4 User_Search.PHP “Search about user”:

This page allows searching about user by typing his ID, its displays the user information and the user's log file.

5.3.4.5 Admin_Algo_List .PHP “List algorithms information”:

This page lists all algorithms information to the administrator. This page is accessed when the user clicks on the (All Algorithms) key in the (Admin_Page.PHP).

5.3.4.6 Algo_Search.PHP “Search about algorithm”:

This page allows searching about algorithm by typing its number.

5.3.4.7 Add_Algo.PHP “Add algorithm page”:

This page asks the administrator to enter the name, cost and type of the algorithm.

5.3.48 Add_Algorithm_Check.PHP “Add algorithm check page”:

This page performs a set of checking operations on the information of the algorithms that are entered by the administrator to check its validity.

5.3.49 Alg_Upload.PHP “Uploads algorithm files”:

This PHP file uploads the file necessary for the algorithm from the administrator computer to the server and puts them in specific folder according to the algorithm name and number.

Note: All these pages can be found in the directory (C:\apache\htdocs\project).

Chapter Six

Testing

6.1 Introduction:

Testing is one of the most significant steps of software engineering. With testing we can put the system after all the work has been done under spot to see if it is good as the expectations, and to find out the mistakes, problems or other unexpected emergent and also to see if the system meets the wanted task..

The testing will be according to the schedule time which takes three weeks. It will be as follows.

6.2 Testing Scheduling:

The table below shows the testing time scheduling of the system.

Table 6.1 Testing scheduling.

Estimated Time	The 1 ST Week	The 2 nd Week	The 3 rd Week
Testing Process			
Unit Testing			
Modular Testing			
Sub-System Testing			
System Testing			
Acceptance Testing			

6.3 Testing Procedures:

The system testing procedures are shown in the table 6.2.

Table 6.2 Testing procedures.

Unit Testing:

To implement such testing we have to test unit by unit, so we take each task of the project; that is we take the image and sound processing algorithms and test the work of these algorithms for each case of the project and already work, test all links in the system and be sure that it work well, test the connection with the database successfully implemented, test the connection with the Matlab work in good way, tests the connection with the mail server done in good way and tests the work of the web server and ensure that its work well.

Module Testing:

This type of testing needs to take each algorithm by its own that is the algorithms of image processing did work and so the sound processing algorithm did. The function of Matlab web server and mail server work as has been expected and the function of Mysql database works properly. Test the error detection part of the system work in good way by displaying the needed message to the user if any error occurs.

Sub-System Testing:

This testing aims to take an attitude of testing and in our project there are four main attitudes fist is the Matlab and algorithms, the second attitude is for the web server and Mysql database, the third is the mail server part, and the fourth part is the uploading process of the files form the client to the server. And so the four directions of work did fulfill the work being stated before.

System Testing:

Test the work of the system, that it should be tested with a complete process of creating user -> login to the system -> selecting algorithms -> uploading user file -> process the file -> downloading and routing the result.

When we implemented this process we did discover that the system works very well.

Acceptance Testing :

Here we have to see if the system meets the requirement that the system was dedicated for and at this point the system did meet the task of the system needed.

6.4 Testing Strategies:

This part of the documentation shows the system testing strategies:

6.4.1 Black Box Testing:

This type of testing depends on the work of suggesting a specific problem and stating the system and a proceeding answer for the behavior of the system and then applying these states on the system truly.

Table 6.3 Black Box Testing.

Inputs	Expected Values	Actual Values	Notes
Invalid user registration information.	The system will reject this information.	The system does not accept this information.	Display error message to the user.
Invalid user login information.	The system will mark this user as unauthorized user.	Not allow the user to access the system.	Display error login message to the user.
Invalid user bank card information.	The system will stop this user to work with the algorithms.	Not allow the user to use the system algorithms.	Display bank error message to the user.
Invalid users file type.	The system will not able to upload the file.	Not upload the file.	Not allow the user to continue the operation.
Sending message has the result to unknown email.	Loss of the message.	Fail to send the mail.	None
Error in the user file.	File Processing failed.	Un able to process the file.	None

6.4.2 White Box Testing:

This testing will be done according to the critical paths on the flow chart that we will suspect the value and then see the real value of that critical path:

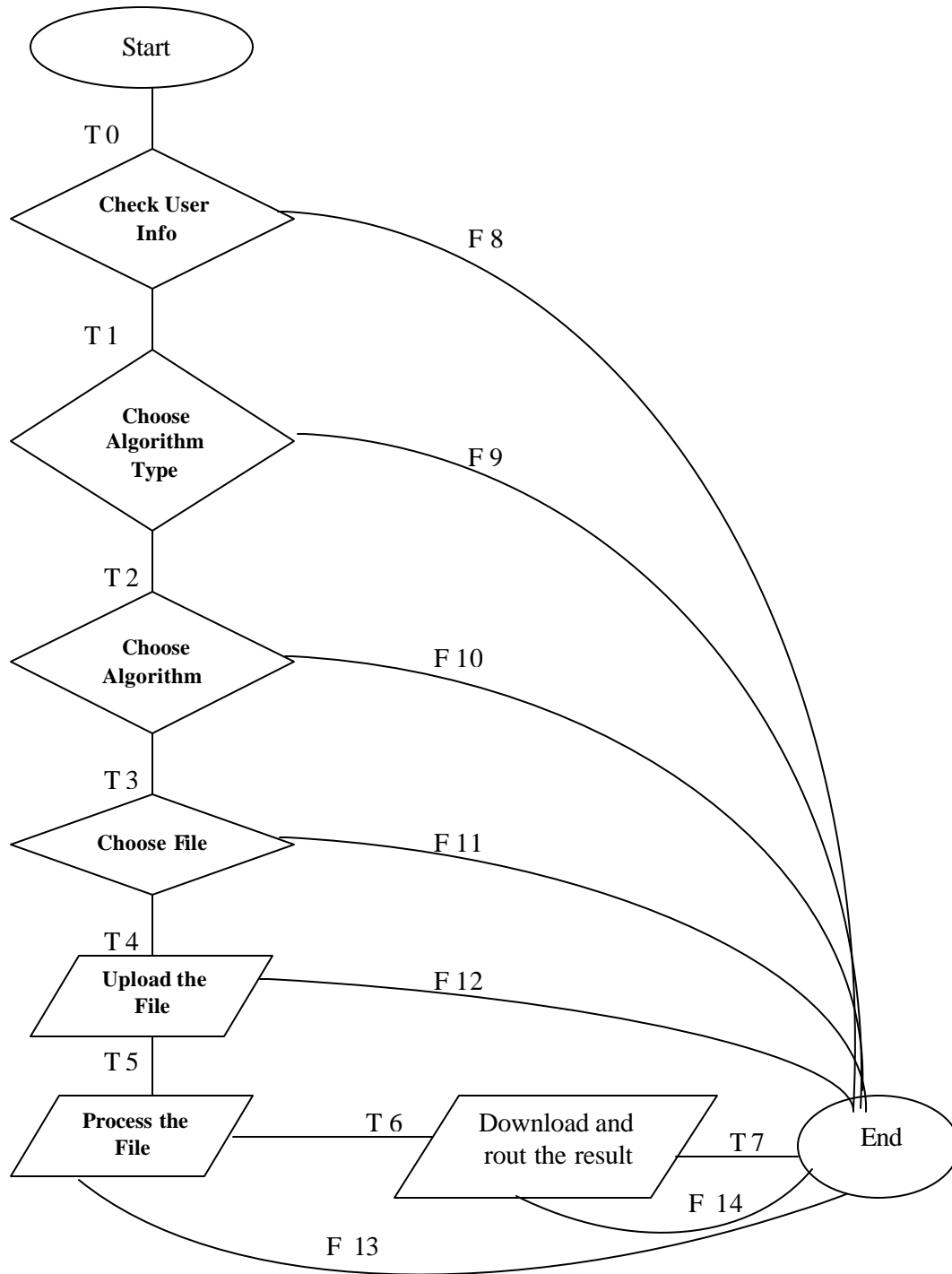


Figure 6.1 White Box Testing

Table 6.4 White box testing.

Branch	Input Value	Expected Value	Actual Value	Notes
1,2,3,4,5,6,7	The user will use the system from the beginning until sending.	The cycle will be performed fluently.	The cycle was ended successfully.	None
8	The user with bad registry number.	The user will not use the system	The system didn't respond.	None
1,9	The user with a valid registry number but haven't chosen algorithm type.	The system will not enable user to continue using the system.	The system waits until choosing algorithm type.	None
1,2,10	A user with valid number and valid algorithm type but haven't chosen algorithm.	The system will not enable user to continue using the system.	The system waiting until choosing algorithm.	None
1,2,3,11	Algorithm information correct but haven't chosen file to process.	Wait until choosing the file.	Wait until choosing the file.	None
1,2,3,4,12	Uploading the File failed.	Stop the process, not enable user to continue.	Stop until the user uploads file successfully completed.	None
1,2,3,4,5,13	Error in Processing the File.	Stop the user operation.	Not enable the user to continue.	None
1,2,3,4,5,6,14	Email name error.	Send result to unknown side.	Send result to unknown side.	None

Chapter Seven

Conclusion and Future Work

7.1 Conclusion:

At the end of this project a number of points can be identified:

1. Working on the problem of building the image and sound processing algorithms gives the ability to understand carefully the meaning and the need for such algorithms.
2. Work on this project increases the knowledge about networking concepts.
3. Working on this project increases the security concepts.
4. This project opens the door to build a more sophisticated and complex project built on Matlab.
5. At the end of this project we have achieved good understanding about the Matlab programming, work with web servers, work with mail servers, send and receive files via Internet, work with database via Internet and web programming applications.

7.2 Future Work:

In this project there are good ideas to work with such techniques that are new, efficient and meet world wide needs.

DSP center is one of the latest applications about connecting the Matlab with the web and this is a very interesting point to the engineers. Additionally, the other components added to the system increase the system effectively and the ability for a practical implementation of such system.

Such system can be improved and rebuilt to give more and more processing filed and can be used as a good technique for:

1. This project can be expanded and changed to work with the medical, financial, control application, broadcast application, etc....
2. This project can be used to implement educational examples and tutorial about many courses in the universities especially signal courses.
3. This system opens the door to operate any Matlab application via Internet and in manageable way.
4. Use Matlab futures with some language that not support or difficult to implement many features like software remote controlling, hardware mechanical controlling, recording voice, etc.....

References

Matlab Information References.

[1] Matlab 6.5 documentation

[2] www.mathworks.com

PHP Information References.

[3] Teach Yourself PHP4 in 24 Hours, Matt Zandstra, <http://www.corrosive.co.uk>,
matt@corrosive.co.uk

[4] PHP HOW-TO, Alavoor Vasudevan, alavoor@yahoo.com

[5] A Programmer's Introduction to PHP 4.0, William Jason Gilmore, info@apress.com,
ISBN # 1-893115-85-2

[6] PHP manual, Alexander Aulbach and many, <http://www.opencontent.org/openpub/>

MYSQL Information References.

[7] <http://www.mysql.com/documentation/>

[8] Web Database Step-by-Step Guide, Ying Zhang, www.sfu.ca/~yzhang/linux/webdb,
yzhang@sfu.ca

[9] Teach Yourself MySQL in 21 Days, Mark Maslakowski

PHP and MYSQL Information References.

[10] Build Your Own Database Driven Website Using PHP and MySQL, Kevin Yank's,
www.sitepoint.com

[11] PHP-And-MySQL Tutorial, Graeme Merrall,
<http://hotwired.lycos.com/webmonkey/programming/php/tutorials/tutorial4.html>

Appendices

These parts of appendices contain the code for the most important pages of the project, to get the complete code for all pages please back to the university library.

Appendix A

Matweb.conf Configuration

This file contains three parts for each algorithm; the first part is the algorithm name, the second is the name of the server (Localhost) and the final part is the path of the algorithm folder.

The (Matweb.conf) file can be edit using windows notepad or by any other text editor software.

```
[Alg1]
mlserver=127.0.0.1
mldir=C:\apache\htdocs\project\Algorithms
```

```
[Alg2]
mlserver=127.0.0.1
mldir=C:\apache\htdocs\project\Algorithms
```

```
[Alg3]
mlserver=127.0.0.1
mldir=C:\apache\htdocs\project\Algorithms
```

.....

Note: That when adding new algorithm the system automatically add field in this file to point to the new algorithm.

Appendix B

(My_Upload_IMAGE.PHP) Code

```
<?
session_start();
session_register("correct_file_submit");
session_register("Output_ext");
?>
```

To indicate that the file successfully uploaded.

```
<?
$path_parts = pathinfo($file);
$file_name=$User_Directory."/";
$file_name.=$Algo_name;
$file_name.="_temp1.";
$file_name.=$path_parts["extension"];
```

To calculate the uploaded file name and destination

```
if( !strcasecmp ($path_parts["extension"],"jpg") ||
!strcasecmp($path_parts["extension"],"jpeg") || !strcasecmp
($path_parts["extension"],"bmp"))
if($path_parts["dirname"]!="")
if ( copy($file, $file_name) )
{
```

Test the validity of the file type the file .

Upload the file .

```
$correct_file_submit=1;
    $Output_ext=".".$path_parts["extension"];
```

To indicate that the upload process successfully completed.

```
}
else
{
    $correct_file_submit=0;
}
```

Call alg#_intro.php page.

```
    $page_name="/Alg$AlgNumber/intro_Alg";
$page_name.=$AlgNumber;
$page_name=".php ";
require("$page_name");
```

```
?>
```

Appendix C

(My_Upload_SOUND.PHP) Code

```
<?
session_start();
session_register("correct_file_submit");
session_register("Output_ext");
?>
```

To indicate that the file successfully uploaded.

```
<?
$path_parts = pathinfo($file);
$file_name=$User_Directory."/";
$file_name.=$Algo_name;
$file_name.="_temp1.";
    $file_name.=$path_parts["extension"];
```

To calculate the uploaded file name and destination

```
if( !strcasecmp($path_parts["extension"],"wav") ||
!strcasecmp($path_parts["extension"],"mp3" ) )
    if($path_parts["dirname"]!="")
        if ( copy($file, $file_name) )
```

Test the validity of the file type the file.

```
{
$correct_file_submit=1;
    $Output_ext=".".$path_parts["extension"];
```

Upload the file.

```
}
else {
```

To indicate that the upload process successfully completed.

```
    $correct_file_submit=0;
```

```
}
```

```
$page_name="/Alg$AlgNumber/intro_Algorithm";
$page_name.=$AlgNumber;
$page_name=".php";
require($page_name);
```

Call alg#_intro.php page.

```
?>
```

Appendix D

(Send_Result_To.PHP) Code

```

<?
if($sended!=1)
{
    session_start();

    $Result_Name="C:/apache/htdocs/project/USER_LOGIN/$Output_Name/".$O
    utput_Name.$Output_ext;

    $fileatt = $Result_Name; // Path to the file
        $fileatt_type = "application/octet-stream"; // File Type
        $fileatt_name = "Result"; // Filename that will be used for the file as the
    attachment

    $email_from = "Service@DSP_Center"; // Who the email is from

    $email_subject = "Result"; // The Subject of the email
    $email_message = "This Is Processing Result Sent By Digital Signal
    Processing Centre;". "\n\n"; // Message that the email has in it
        $email_message .= "This email contains File Sented to you by
    $User_Name.";
    $headers = "From: $email_from";

    $semi_rand = md5(time());
    $mime_boundary = "=="Multipart_Boundary_x{$semi_rand}x";

    $headers .= "\nMIME-Version: 1.0\n" .
        "Content-Type: multipart/mixed;\n" .
        " boundary=\\"{$mime_boundary}\\"";
    $email_message .= "This is a multi-part message in MIME format\n\n" .
        "--{$mime_boundary}\n" .
        "Content-Type:text/html; charset=\\"iso-8859-1\\"\n" .
        "Content-Transfer-Encoding: 7bit\n\n" .
    $email_message . "\n\n";

    $file = fopen($fileatt,'rb');
    $data = fread($file,filesize($fileatt));
    fclose($file);

    $data = chunk_split(base64_encode($data));

    $email_message .="--{$mime_boundary}\n"."Content-Type:
    {$fileatt_type};\n"." name=\\"{$fileatt_name}\\".\n"."Content- Transfer-Encoding:
    base64\n\n".$data. "\n\n"."--{$mime_boundary}--\n";

```

Send the email if it is not sent before.

Result full name.

To calculate the parameters needed for sending emails.

Configuration the email header and the message body.

```
/** ***** End of File Config ***** */
```

```
// To add more files just copy the file section again, but make sure they are all  
// one after the other! If they are not it will not work!
```

```
if($send_email1=="send ")
```

```
{  
$email_to = $send_email_to1;
```

```
@mail($email_to, $email_subject, $email_message, $headers);  
$sended=1;
```

```
}
```

```
if($send_email2=="send ")
```

```
{  
$email_to = $send_email_to2;
```

```
@mail($email_to, $email_subject, $email_message, $headers);  
$sended=1;
```

```
}
```

```
}
```

```
?>
```

Send the email.

Appendix E

(Alg1.m) Code

(Example of image processing algorithm)

```
function retstr = Alg1(instruct,outstruc)
retstr = char("");
root=instruct.root;
Name=instruct.Out_Name;
Output_ext=instruct.Output_ext;
file_name=instruct.Algo_name;
```

Input parameter taken from the system

```
file_name=[file_name,'_temp1'];
file_name=[file_name,instruct.Output_ext];
```

Results full name.

```
Zip_name = [Name, '.zip'];
full_out_name = [Name, Output_ext];
cd(root);
P=imread(file_name);
ED = edge(rgb2gray(P),'Roberts',0.05);% "Roberts" method w/ threshold
imwrite(ED,full_out_name);
```

Process the image file using "Roberts" method.

```
zip(Zip_name,full_out_name);
```

Make a compressed version of the result.

```
outstruct.GraphFileName = sprintf(Name);
```

```
phpfile_name=[instruct.Algo_name,'.php'];% Alg#.php
templatefile = which(phpfile_name);
if (nargin == 1)
retstr = htmlrep(outstruct,templatefile);
elseif (nargin == 2)
retstr = htmlrep(outstruct,templatefile, outfile);
end
```

Output the result to the web part of the system.

Appendix F

(Alg7.m) Code.

(Example of sound processing algorithm)

```
function retstr = Alg7(instruct,outstruct)
retstr = char("");
root=instruct.root;
Name=instruct.Out_Name;
Output_ext=instruct.Output_ext;
file_name= instruct.Algo_name;
SORH=instruct.SORH;
SCAL=instruct.SCAL;
N=instruct.Num;
```

Input parameter taken from the system.

```
file_name=[file_name,'_temp1'];
file_name=[file_name,instruct.Output_ext];
```

Results full name.

```
Zip_name = [Name, '.zip'];
full_out_name = [Name, Output_ext];
cd(root);
```

Process the image file using "sqtwolog" method.

```
lev = str2num(N);
xd = wden(file_name,'sqtwolog',SORH,SCAL,lev,'sym8'); %'sqtwolog' for universal threshold sqrt(2*log(.)).
```

```
WAVWRITE(xd,full_out_name); %de noise signal
```

Make a compressed version of the result.

```
zip(Zip_name,full_out_name);
```

```
outstruct.GraphFileName = sprintf(Name);
```

Output the result to the web part of the system.

```
phpfile_name=[instruct.Algo_name,'.php'];% Alg#.php
templatefile = which(phpfile_name);
if (nargin == 1)
retstr = htmlrep(outstruct,templatefile );
elseif (nargin == 2)
retstr = htmlrep(outstruct,templatefile, outfile);
end
```

Appendix G

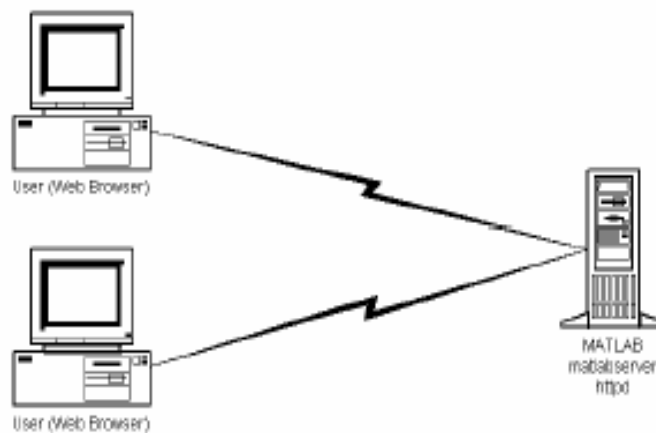
Who to Use Matlab Web Server

Introduction

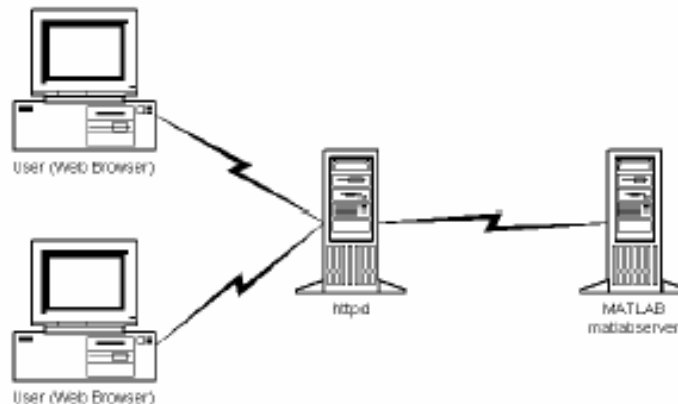
MATLAB Web Server Environment

The MATLAB[®] Web Server enables you to create MATLAB applications that use the capabilities of the World Wide Web to send data to MATLAB for computation and to display the results in a Web browser. The MATLAB Web Server depends upon TCP/IP networking for transmission of data between the client system and MATLAB. The required networking software and hardware must be installed on your system prior to using the MATLAB Web Server.

In the simplest configuration, a Web browser runs on your client workstation, while MATLAB, the MATLAB Web Server (`matlabserver`), and the Web server daemon (`httpd`) run on another machine.



In a more complex network, the Web server daemon can run on a machine apart from the others.



Building MATLAB Web Server Applications

MATLAB Web Server applications are a combination of M-files, Hypertext Markup Language (HTML), and graphics. Knowledge of MATLAB programming and basic HTML are the only requirements.

The application development process requires a small number of simple steps:

- 1 Create the HTML documents for collection of the input data from users and display of output. You can code the input documents using a text editor to input HTML directly, or you can use one of the commercially available HTML authoring systems, such as Front Page from Microsoft, PageMill from Adobe, or HoTMetaL from SoftQuad.
- 2 List the application name and associated configuration data in the configuration file `matweb.conf`. (See "matweb.conf" for a description of this file.)
- 3 Write a MATLAB M-file that:
 - a Receives the data entered in the HTML input form.
 - b Analyzes the data and generates any requested graphics.
 - c Places the output data into a MATLAB structure.
 - d Calls `htmlrep` to place the output data into an HTML output document template. (See `htmlrep` for a description of this process.) The maximum amount of HTML data you can receive from MATLAB is 256 KB.

Product Requirements

The MATLAB Web Server has the same supporting hardware and software requirements as MATLAB itself, except for memory required. MATLAB hardware and software requirements are documented in the MATLAB Installation Guide for your computer.

Memory requirements while running the MATLAB Web Server vary with the number of MATLABs configured. Each MATLAB running under the MATLAB Web Server consumes 256 KB of memory.

The MATLAB Web Server requires that TCP/IP networking software must be installed on your computer.

Web Requirements

Web Browsers

To submit input to and receive output from the MATLAB Web Server, you must install a Web browser suitable for your platform. Current versions of the MATLAB Web Server have been tested with Netscape Communicator Release 4.7 and Microsoft Internet Explorer 5.0. Earlier browser releases will probably also work but have not been tested.

The MathWorks does not redistribute these products. You can obtain them directly from the companies that developed them. You can find additional information at <http://www.netscape.com> or <http://www.microsoft.com>.

Web Server

You need to install Web server software (HTTPD or similar) on the system where MATLAB is running or on a machine that has network access to the machine where MATLAB is running. There are numerous sources for obtaining this software, including:

- Pre-installed Microsoft Peer Networking Services on your PC
- Netscape Enterprise Server, available by purchase from Netscape Communications, Inc.
- Free distribution over the Internet (Apache: <http://www.apache.org>)

The Web server software must be capable of running Common Gateway Interface (CGI) programs.

Installation

Availability

The MATLAB Web Server is available on UNIX (Solaris) workstations and IBM PC compatible computers running Microsoft Windows NT or Linux.

Installation Procedure

To install the MATLAB Web Server, follow the normal MATLAB installation procedure for your platform, as documented in the *MATLAB Installation Guide for PC* and the *MATLAB Installation Guide for UNIX*. The MATLAB Web Server appears as one of the installation choices you can select as you proceed through the installation screens.

General Post-Installation Procedures

After installing MATLAB and the MATLAB Web Server, you must perform a number of steps that regulate communication between MATLAB and your Web browser.

Note Throughout this document the notation `<matlab>` represents the MATLAB root directory, the directory where MATLAB is installed on your system. For notational consistency in UNIX command syntax, the notation `$MATLAB` is used to represent the MATLAB root directory.

Note in particular:

- 1 To get the demonstration programs discussed in this chapter to work, you need to create a `matweb.conf` file in the `<matlab>/toolbox/webserver/wsdemos` directory. The `Readme` file shows the format for `matweb.conf`. Replace the notation `<matlab>` with the name of the root directory where you installed MATLAB. Use the `matlabroot` command to determine this directory. Also, replace `matlabserver_host_name` with the TCP/IP hostname for your machine.
- 2 The installation procedure creates the file `matlabserver.conf` in the `<matlab>/webserver` directory. The file contains the notation

- 1

This number represents the number of MATLABs that can run concurrently. After testing that everything is working properly, you can change this number to something more convenient. On Windows NT edit `matlabserver.conf` directly with a text editor.

Note To see any changes made to `matlabserver.conf`, you need to stop and restart the MATLAB Web Server. See the section "Understanding `matlabserver`" on page 3-5 for more information on `matlabserver.conf`.

On Solaris/Linux use the `webconf` script to initialize the `matlabserver.conf` file. Then edit this file to change options further, particularly those that `webconf` cannot set on the command line.

3 Follow the directions provided by your Web server (`httpd`) to create the needed aliases:

- a** The home or default directory
- b** `/cgi-bin`
- c** `/icons`

Point each of these aliases to `<matlab>/toolbox/webserver/wsdemos` to get the demonstration programs to work.

If your application creates graphic (jpeg) files, you need to provide a location where MATLAB can write these for `httpd` access, e.g. `/icons`. The `mkdir` entry associated with each application in the `matweb.conf` file indicates the location to MATLAB.

If you do not have permission to set up or change these aliases, you must place copies of some files in locations where the `httpd` can find them.

- Copy `matweb` (`matweb.exe` on Windows NT), found in `<matlab>/webserver/bin/arch`, to the directory aliased by `/cgi-bin` or equivalent. The supported architectures are Windows NT (`win32`), Solaris (`sol2`), and Linux (`glnx86`).
- Copy `matweb.conf` in `<matlab>/toolbox/webserver/wsdemos` to the directory aliased by `/cgi-bin` or equivalent.
- Copy all demo HTML files in `<matlab>/toolbox/webserver/wsdemos` to the directory where the `httpd` keeps all HTML files (often referred to as the *home or default* alias).

Note that when aliases are different from those provided in the demo HTML files, you will have to make the corresponding changes in those HTML files. For example, if you use the Apache Web Server (<http://www.apache.org>), the aliases listed above are called:

- a** `DocumentRoot`
- b** `ScriptAlias /cgi-bin/`
- c** `Alias /icons/`

You edit these in the `conf/http.conf` file in the main Apache directory. Note that any alias with a trailing `'/'`, such as `'/icons/'`, *must* have a trailing slash in its value, e.g.,

```
Alias /icons/ "/local/matlab/toolbox/webserver/wsdemos/"
```

Windows NT Post-Installation Procedures

After installation, you must reboot your machine to start **MATLAB Server** as a Windows NT service. The service starts automatically at system boot.

Startup Sequence

If you install a new version of MATLAB and the MATLAB Web Server, start MATLAB before starting Web Server. MATLAB performs some system updates required for successful Web Server operation.

Deinstallation

To remove **MATLAB Server** from the Windows NT Registry, open a command prompt (MS-DOS) window. Enter the command sequence

```
cd <matlab>/webserver/bin/win32
matlabserver -remove
```

Introduction

The process of creating a MATLAB Web Server application involves the creation of:

- An HTML input document for data submission to MATLAB. See "Creating Input Documents" on page 2-4.
- An HTML output document for display of MATLAB's computations. See "Creating Output Documents" on page 2-10.
- A MATLAB M-file to process input data and compute results. See "Creating MATLAB Web Server M-Files" on page 2-7.
- A test file to validate code before distributing the application over the Web. See "Debugging Your Application" on page 2-13.

The process of creating a MATLAB Web Server application can be simplified through the use of a set of templates that has been provided. These are discussed in "Templates" on page 2-2.

Templates

Four templates found in the directory `<matlab>/toolbox/webserver/wsdemos` simplify the process of creating a MATLAB Web Server application:

- `input_template.html`
- `output_template.html`
- `mfile_template.m`
- `tmfile_template.m`

Each template provides actual code that you need to incorporate into your application plus instructions on how to modify the template where necessary. If you follow the directions in these templates, you should be able to create MATLAB Web Server applications with reasonable effort.

Additionally provided in `<matlab>/toolbox/webserver/wsdemos` is `webmagic`, a magic squares demonstration program. A magic square produces the same sum along any row, column, or either of the two main matrix diagonals. There are four files associated with `webmagic`:

- webmagic1.html: the webmagic input document
- webmagic2.html: the webmagic output document
- webmagic.m: the webmagic MATLAB M-file
- twebmagic.m: the webmagic stand-alone test file

To learn how the four templates were modified to create the webmagic application, we will examine templates and note the specific changes applied.

If you want to look at some other applications created with these templates, see the section "Additional Application Examples" on page 2-15.

Creating Input Documents

Input Template

The file `input_template.html` provides the code needed to create a MATLAB Web Server input document. An abbreviated version looks like

```
<!-- STEP 1
Choose either the NT version or the Unix version of the form
tag (depending on which platform the matweb client program
will be run):
-->
<!-- NT version: -->
<form action="/cgi-bin/matweb.exe" method="POST">
<!-- Unix version: -->
<form action="/cgi-bin/matweb" method="POST">

<!-- STEP 2
Create a hidden field naming your M-file. Replace MY_M_FILE
with the name of main MATLAB function of your application. (An
HTML input field of type 'hidden' is commonly used to pass
variables to a web server. It is not displayed by the
browser.)
-->
<input type="hidden" name="mlmfile" value="my_m_file">

<!-- STEP 3
Add all your other HTML form tags here. Replace
MY_INPUT_VARIABLE_1 with the name of an input variable in
your application.
-->
<p>My input variable 1: <input type="text"
name="my_input_variable_1">
<!--
Create additional input variables here.
-->

<!-- STEP 4
Create a 'submit' input tag for the user to click to send
the input to your program.
-->
```

```
<p><input type="submit" name="Submit" value="Submit"></p>
<!-- STEP 5
Add the name of your main application function to the file
matweb.conf. See the matweb.conf file in the wsdemos
directory and the documentation.)
-->
```

webmagic Input

Examine the significant part of the source for webmagic1.html.

```
<form action="/cgi-bin/matweb.exe" method="POST">
  <input type="hidden" name="mlmfile" value="webmagic">
  <p>Magic square size (minimum is 3, maximum is 20):
  <input type="text" size="2" maxlength="2"
name="msize"></p>
  <p><input type="submit" name="Submit" value="Submit"></
p>
</form>
```

The line

```
<form action="/cgi-bin/matweb.exe" method="POST">
```

calls `matweb`, the entry point to the MATLAB Web Server. `matweb.exe` is the Microsoft Windows NT name of the program used by the MATLAB Web Server to extract data from HTML forms. On Solaris/Linux this program is called just `matweb`. We refer to the program as `matweb` throughout this document except when the platform distinction is important. `matweb` is described more thoroughly in the next chapter.

The next line

```
<input type="hidden" name="mlmfile" value="webmagic">
```

provides the name of the MATLAB M-file (`mlmfile`) to run. In this application the M-file is named `webmagic`.

Lastly, the input

```
<input type="text" size="2" maxlength="2" name="msize"></p>
```

passes to `webmagic.m` a two-character field named `msize`, which contains the size of the magic square to compute.

To display the input document and run the magic squares demonstration locally on your computer, start your Web browser and set the URL to `http://<your_domain>/webmagic1.html`.

The magic squares input document, `webmagic1.html`, is displayed in your browser.



The screenshot shows a web page with the following content:

- The MATH WORKS Inc.** logo with the tagline "Developers of MATLAB".
- The heading "Display a MATLAB Matrix in an HTML Table".
- A text input field labeled "Magic square size (minimum is 3, maximum is 20):" with a small square icon to its right.
- A "Submit" button.
- A footer line: "©1990 by The MathWorks, Inc. All rights reserved. MATLAB is a registered trademark of The MathWorks, Inc."

Enter the size of the magic square matrix you want to compute and press **Submit**.

Creating MATLAB Web Server M-Files

M-File Template

You use the M-file template to code your MATLAB application as you normally do. The template provides the additional code you need to accept input from your HTML input document and to return results to your HTML output document. An abbreviated version looks like

```
function retstr = mfile_template(instruct, outfile)
% STEP 1
% Initialize the return string.
retstr = char('');

% STEP 2
% Set working directory.
% The variables INSTRUCT.MLDIR and INSTRUCT.MLID are
% provided
% automatically to all MATLAB Web Server applications that
% use
% the matweb program.
cd(instruct.mldir);

% STEP 3
% Get the HTML form input variables
my_input_variable_1 = instruct.my_input_variable_1;

% STEP 4
% Perform your MATLAB computations, graphics file creations,
% etc., here:

% STEP 5
% Put variables that you want to put into your HTML output
% document in an output structure. You create an HTML output
% document from OUTPUT_TEMPLATE.HTML.
outstruct.my_output_variable_1 = More MATLAB computations
creating ...
    scalars, matrices, cell arrays, graphics files, etc.;

% STEP 6
% Call the function HTMLREP with the output structure you
% just
```

```

% created and the filename you created from
OUTPUT_TEMPLATE.HTML.
% Replace <OUTPUT_TEMPLATE.HTML> with the name of the HTML
output
% file you created using OUTPUT_TEMPLATE.HTML.
% This call fills the string RETSTR to return and optionally
% writes the output as a file if a valid filename is given
as the
% second argument to the present function.
templatefile = which('<OUTPUT_TEMPLATE.HTML>');
if (nargin == 1)
    retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
    retstr = htmlrep(outstruct, templatefile, outfile);
end

```

webmagic M-File

The data entered on the webmagic1.html input document is automatically passed to MATLAB, which then runs the webmagic function. Notations in **boldface** refer to steps in the M-file template.

```

% Initialize the return string. (Step 1)
retstr = char('');

(Step 2. Not needed. No generated graphics.)

% Get the msize (string) variable. Convert to a number. (Step 3)
% Check the range.
if (~length(instruct.msize))
    msize = 3; % Default empty field.
else
    msize = str2double(instruct.msize);
    if (msize > 20), msize = 20; end % Max size.
    if (msize < 3), msize = 3; end % Min square.
end

% Save size as a char string in structure OUTSTRUCT. (Step 4, 5)
outstruct.msize = msize;

% Create magic square in output structure OUTSTRUCT.
outstruct.msquare = magic(msize);

% Get column, row, and diagonal sum. Put in OUTSTRUCT.
d = sum(outstruct.msquare,1);
outstruct.msum = d(1,1);

% Output the results and optionally write as a file if the
% filename was given as the second argument to WEBMAGIC.
(Step 6)
templatefile = which('webmagic2.html');
if (nargin == 1)
    retstr = htmlrep(outstruct, templatefile);
elseif (nargin == 2)
    retstr = htmlrep(outstruct, templatefile, outfile);
end

```

Creating Output Documents

Output Template

The file `output_template.html` provides the code needed to create a MATLAB Web Server output document. An abbreviated version looks like

```
<!--  
Modify this file to create your own HTML output document  
and save it as <MY_OUTPUT>.html, where <MY_OUTPUT> is  
replaced  
by a name that has meaning within the context of your  
application.  
-->  
  
<!-- STEP 1  
Display a MATLAB scalar or character string. Replace  
<MY_OUTPUT_VARIABLE_1> in the following line with the name  
of the MATLAB variable you want to display. Change the other  
text to something meaningful within the context of your  
application.  
-->  
My output variable 1 has been computed to be  
$<my_output_variable_1>$  
  
<!-- STEP 2  
Put all your other HTML tags here.  
-->
```

webmagic Output

The `webmagic` output document contains three variables.

`$msquare$` -- the completed magic square

`$msize$` -- the size of the magic square

`$msum$` -- the magic square sum along its rows, columns, or diagonals

Using `htmlrep` the `webmagic` function replaces these variables with actual values, using the input obtained from `webmagic1.html`.

The source for the webmagic2.html template document looks like

```
<!--
HTML output template used by webmagic.m in call to the
function HTMLREP. (HTMLREP replaces variable names delimited
by dollar signs, e.g., $msquare$, with their values. It also
generates HTML tables and select lists dynamically from
matrices,
cell arrays, and vectors.)
-->

<html>
<head>
<title>Magic Square in an HTML Table</title>
</head>
<div align='center'>
<strong>Magic Square in an HTML Table</strong></p>

<!--
Use the MATLAB "AUTOGENERATE" HTML attribute to generate a
table dynamically from the variable 'msquare', which is a
matrix (in program webmagic.m).
-->

<table border="1" cellspacing="1" autogenerate='$msquare$'>
  <tr>
    <td align='right'>
  </td>
</tr>
</table>

<!--
Replace $msize$ and $msum$ with the contents of MATLAB
variables
named 'msize' and 'msum' respectively as computed in
webmagic.m.
-->

<p>
Size: $msize$<br>
Sum of all columns, rows, and diagonals: $msum$
</div>
```

When displayed in a browser, webmagic2.html looks like



MATH BY
MATLAB

Magic Square in an HTML Table

10	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

Size: 4
Sum of all columns, rows, and diagonals: 34

©1999 by The MathWorks, Inc. All rights reserved. MATLAB is a registered trademark of The MathWorks, Inc.

Appendix H

Use and Install Apache Web Server

If you ignore this text, you WILL be told to refer back to it if you ask a question in the PHPGeek forums. How nicely will depend on the respondent

This installer sets up a PHP server environment on your Windows machine. This software is provided under a variety of free software licenses. All of them allow you to freely redistribute this software as you received it. They do NOT all allow commercial redistribution. This installer only installs the core applications. There are various modules available to extend the functionality at phpgeek.com. Please direct any support questions first to the FAQ at phpgeek.com and second to the support forums at that same site. Due to the high volume of email I receive, I do NOT provide any support by email and any support requests by email will be deleted without a response

Also, keep in mind that while this software is free, it does consume time, energy and financial resources to develop and support it. Please consider making a donation or buying some of the branded merchandise available at phpgeek.com

GETTING STARTED

The software will be installed to `c:\apache`. Under that directory are all of the components including Apache, PHP and MySQL. Apache and MySQL are servers that need to be started in order for them to function. They operate in the background to do their jobs. There are shortcuts in the PHPTriad folder in your start menu to start both. Once Apache is started, it will remain open in a DOS window. MySQL will flash a DOS window and go away. You can check to see if MySQL is running by doing a CTRL-ALT-DEL and looking for a `mysqld*` process running

Once both servers are started, go to `http://localhost` or `http://127.0.0.1`. These are the loopback addresses for your local computer and are the addresses you can use to test your PHP applications. `http://localhost` is mapped to `c:\apache\htdocs`. Individual files in that directory will be available at `http://localhost/filename.ext`. Any directories you create under `c:\apache\htdocs` will also be reflected in the URL as well. `c:\apache\htdocs\directory` will be accessed at `http://localhost/directory`

There are more details available when you load `http://localhost` for the first time