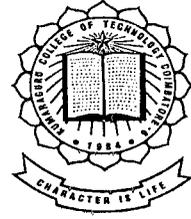


P-1726



WEB ADMINISTRATION CONSOLE

By

D. KOUSALYA DEVI

Reg. No 71203621022

of

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore

(Affiliated to Anna University)

A PROJECT REPORT

Submitted to the

FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING

In partial fulfillment of the requirements

for the award of the degree

of

MASTER OF COMPUTER APPLICATIONS

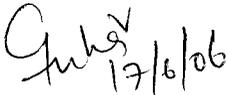
June, 2006

KUMARAGURU COLLEGE OF TECHNOLOGY

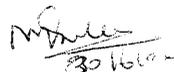
Coimbatore – 641006

BONAFIDE CERTIFICATE

Certified that this project report titled **WEB ADMINISTRATION CONSOLE** is the bonafide work of **Ms. D. KOUSALYA DEVI (Reg No. 71203621022)** who carried out the research under my supervision .Certified further, that to the best of my knowledge the work reported here in does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PROJECT GUIDE****HEAD OF THE DEPARTMENT**

Submitted for the University Examination Held on 30.06.06

**INTERNAL EXAMINER****EXTERNAL EXAMINER**

ÜG

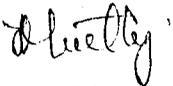
Valluvar Kottam High Road,
ambakkam, Chennai - 600 034.
44 52187905.

KLUG
INTELLIGENT IT SOLUTIONS

Project Completion Certificate

This is to certify that **Ms. D.KOUSALYA DEVI**,
Reg no 71203621022, final year M.C.A, Kumaraguru College Of Technology
Coimbatore. has completed her project in this organization. She did the
project entitled "**WEB ADMINISTRATION CONSOLE** " in Dot Net from
January 2006 to June 2006 at our organization in partial fulfillment of the
requirements for the award of M.C.A. Anna University. Her work was timely
from the company's perception and useful to the company.

As part of the company's policy we don't let out any
sort of coding or pseudo code out of the company's premises in printed or in
electronic media.



(HR - Manager)

ABSTRACT

The project entitled "**Web Administration Console**" is used to manage and administer a server located at the remote place. In this methodology of server administration there is no need for a person to be present physically at the server for performing the management activities.

The main objective of the software is to provide a web-based system to manage the server administration work. It includes authentication of groups and users, file uploading and deleting, scheduling the works like error debugging, code manipulations database management etc from a remote location.

There are several modules which communicate with each other. The main functionality of this project is Data Base Management where it can be able to create all database queries, insertion, deletion and updation of tables.

The other modules in this project are group manager, file creation/deletion, directory creation/deletion, code manager, schedule, and authentication and Stegnography.

Administrator has the availability to log into the administration console and access administrative functions through the console. The project has been developed using Visual Basic .NET as the front end with SQL SERVER as the backend.

ACKNOWLEDGEMENT

I wish to express my sincere thanks to **Dr. V. Joseph Thanikal** , Principal, Kumaraguru College of Technology, Coimbatore, for permitting me to undertake this project.

I wish to express my sincere thanks to **Dr. M. Gururajan**, Head , Department of Computer Applications, Kumaraguru College of Technology, Coimbatore, for his timely help and guidance throughout this project.

I also express my thanks to my guide **Mrs.V.Geetha** , Assistant Professor, Department of Computer Applications, Kumaraguru College of Technology, for her valuable guidance and encouragement at every stage of this project work.

I express my sincere thanks to **Mrs. A. Mumtaj Begum**, Project Manager, Klugit, for her support and assistance at various levels of my project work.

I express my sincere thanks to **Mr. V. Vinaytheerthan**, Senior Analyst, Klugit, Chennai for his invaluable guidance support and suggestion throughout the course of this project work.

Last but not the least; I am grateful to my parents and friends for their constant support and encouragement.

TABLE OF CONTENTS

	PAGE NO
ABSTRACT	iii
ACKNOWLEDGEMENT	iv
LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF SYMBOLS, ABBREVIATIONS AND NOMENCLATURE	viii
1. INTRODUCTION	1
1.1 SYSTEM OVERVIEW	1
1.2 OBJECTIVE OF THE PROJECT	2
1.3 ORGANIZATION PROFILE	3
2. SYSTEM REQUIREMENT AND SPECIFICATION	4
2.1 HARDWARE REQUIREMENTS	4
2.2 SOFTWARE REQUIREMENTS	4
2.3 SOFTWARE OVERVIEW	5
3. SYSTEM ANALYSIS	8
3.1 PROBLEM STATEMENT	8
3.2 EXISTING SYSTEM	8
3.2.1 DRAWBACK OF EXISTING SYSTEM	9
3.3 PROPOSED SYSTEM	9
3.3.1 BENEFITS OF PROPOSED SYSTEM	11

4. SYSTEM DESIGN	12
4.1 ELEMENT OF DESIGN	12
4.1.1 INPUT DESIGN	12
4.1.2 OUTPUT DESIGN	13
4.1.3 ARCHITECTURE DESIGN	14
4.1.4 DATABASE DESIGN	14
4.2 TABLE DESIGN	16
5. DATA FLOW DIAGRAM	21
5.1 DATA FLOW DIAGRAM	21
6. SYSTEM TESTING AND IMPLEMENTATION	26
6.1 TEST PLAN	30
6.2 SYSTEM IMPLEMENTATION	27
6.2.1 ALGORITHM IMPLEMENTED	27
7. CONCLUSION AND FUTURE ENHANCEMENT	37
7.1 CONCLUSION	37
7.2 FUTURE WORK	38
APPENDICES	39
REFERENCES	50

LIST OF TABLES

Table No.	Description	Page No.
4.1	Administrative login	16
4.2	Group Master	16
4.3	Schedule	17
4.4	Comment	18
4.5	Users Type	18
4.6	Rights	19
4.7	Users	20
6.1	Test Plan	26

LIST OF FIGURES

Figure No.	Description	Page No.
5.1	Data Flow Diagram Level 0	21
5.2	Data Flow Diagram Level 1	22
5.3	Data Flow Diagram Level 2.1	23
5.4	Data Flow Diagram Level 2.2	24
5.5	Data Flow Diagram Level 2.3	25
4.1	E-R Diagram	14
A1.1 – A1.11	Screen Shots	39

LIST OF SYMBOLS, ABBREVIATIONS, AND DEFINITIONS

Abbreviations	Explanations
ASP	Active Server pages
WSI	Web Service Interface
HTTP	Hyper Text Transfer Protocol
VB	Visual Basic
SQL	Structured Query Language
HTML	Hyper Text Markup Language
IIS	Internet Information Service

CHAPTER 1

INTRODUCTION

1.1 SYSTEM OVERVIEW:

This project entitled "Web Administration Console" is a web based application that can be used for the management and administration of a server and the site associated with it. This web project deals with administration of a server located in a remote location. The server can be configured and administered effectively through this application. The server administration can be done on any website, preferably one involving database manipulation. This server management process can be done effectively from the client side, with no physical presence of the server.

The various tasks are server administration, Group and database management, File/Directory services, scheduling of various activities, Code manipulations and authentication.

The Group manager deals with the new group creation, view existing groups and deleting unnecessary groups.

The D manager deals with the entire database manipulations of the site. This module provides such functionality that all the database operations and table manipulations can be performed, with no necessity to go to the back-end.

With the File and Directory service, we can create files and directories, upload files, and delete files or directories from the client itself.

The source code module is used for creating and editing application or web pages within the site. These operations can be done with ease from the client side also.

The schedules section allows the site manager or administrator to set schedules and works to be done at specific times.

The Administrator section allows the administrator for changing their ID's and passwords required to gain access to the site.

1.2 OBJECTIVE OF THE PROJECT:

This new application has brought an innovative feature in the current methods of web server management. This web-based application enables us to manage or administer a server, located at the remote place.

There is no necessity that one has to be present physically at the server for performing various site management activities. There will be lots of activities that has to be performed with a web site like viewing the site contents, saving the contents etc.,

The application allows one to do all these process from the client itself, with no necessity for present at the server itself. The group manager can perform various operations like adding new users to the group, deleting users already present from the group and viewing the users in the existing groups with ease from a remote place itself.

The system also provides facilities to alter or modify the tables present within the site with no fear of data loss. There is no need to go to backend at all, for these processes. These operations can be performed from the front end, that too from the client side itself. A GUI based interactive environment is provided to create and edit the tables and also to modify the structure of the table, allowed by the database management system.

Another additional feature of this application is Steganography that hides the message but not the fact that two parties are communicating with each other. Algorithm used to perform steganography is Huffman Algorithm. The steganography process generally involves placing a hidden message in some transport medium, called the carrier. The secret message is embedded in the carrier to form the steganography medium. The use of a steganography key may be employed for encryption of the hidden message and/or for randomization in the steganography scheme. In summary:

steganography_medium= hidden_message + carrier +steganographykey

Various outputs are generated within the system. It allows the server manager to know of various status of the system and also about the updating that are taken place within the server.

1.3 ORGANIZATION PROFILE

KLUGIT is a dynamic organization providing *Software development solutions*. It is a software development company specializing projects to the clients based in India and US and in an ongoing commitment to quality, the Company's continuing goal is to enhance its position as one of the leading independent providers of resources for ITES and IT industries.

Klugit is ideally qualified to offer its Clients the best possible solution in fulfilling their technical needs with the professionals having more than 4 years of software development experience, *Klugit* is a dynamic organization providing *Software development solutions*. It is a software development company specializing projects to the clients based in India and US and in an ongoing commitment to quality, the Company's continuing goal is to enhance its position as one of the leading independent providers of resources for ITES and IT industries.

CHAPTER 2

SYSTEM REQUIREMENT AND SPECIFICATION

The system requirement and specification provided here are used in the development of this system.

2.1 HARDWARE REQUIREMENTS

The hardware components on which this application is developed are,

Processor	:	Pentium III
Monitor	:	SVGA
RAM	:	128MB(minimum)
Speed	:	500MHZ
Secondary device	:	20GB
Floppy Disk	:	1.44 inch
Keyboard	:	104 Keys
Mouse	:	Logitech
CD Drive	:	52X Disc Drive

2.2 SOFTWARE REQUIREMENTS

The software components required to develop this application are,

Operating System	:	Windows 98/NT/2000/XP
Framework	:	.Net Framework
Front-End	:	ASP. NET , VB. NET
Back-End	:	SQL-Server 2000

2.3 SOFTWARE OVERVIEW

VB.NET which is the new version of Visual Basic. There is Visual Studio.NET, an Integrated Development Environment that hosts VB.NET, C#, and C++.NET. Underlying all this is the .NET Framework and its core execution engine, the Common Language Runtime. In the .NET model, you write applications that target the .NET Framework. This gives them automatic access to such benefits as garbage collection (which destroys objects and reclaims memory for you), debugging, security services, inheritance, and more. When you compile the code from any language that supports the .NET Framework, it compiles into something called MSIL, or Microsoft Intermediate Language. This MSIL file is binary, but it is not machine code; instead, it is a format that is platform independent and can be placed on any machine running the .NET Framework. Within the .NET Framework is a compiler called the Just-In-Time, or JIT, compiler. It compiles the MSIL down to machine code specific to that hardware and operating system.

ADO.NET

ADO.NET uses a disconnected architecture to operate. The reason for this is that traditional applications that maintained an open database connection did not scale well. If a component opened a connection to the database and held it open for the lifetime of the application, it consumed expensive database resources, when the connection probably was needed for only a small percentage of that time.

As the number of users grows, the overhead of the database connections can begin to affect the database performance negatively. Therefore, Microsoft decided to use a disconnected architecture for ADO.NET. This not only solves the problems with scalability by reducing the number of active connections, it makes it much easier to transfer data from one component to another. You do not need to have both components connected to the database, nor do you have to have them both understand some binary format of the data, as you will see in a moment.

Additionally, Microsoft recognized that in much programming today, you basically have a disconnected application architecture, thanks to the Web. Think of

a typical Web application: The user requests a page and the page is generated on the server, with a mix of HTML and data from the database. By the time the page is sent back to the user and rendered in his browser, the connection to the database is no longer needed. Therefore, if it takes one second to render a page but the user views it for twenty-nine seconds, a database connection is needed for only one-thirtieth of the total time the page is used. If you still need to use a connected architecture, Microsoft recommends you use ADO. ADO.NET is inherently disconnected, so ADO is still a better approach if you need a continuous connection to the underlying database.

Overview of SQL SERVER 2000

A database is similar to a data file where data gets stored.. Like a data file, a database does not present information directly to a user; the user runs an application that accesses data from the database and presents it to the user in an understandable format.

Database systems are more powerful than data files in that data is more highly organized. In a well-designed database, there are no duplicate pieces of data that the user or application must update at the same time. Related pieces of data are grouped together in a single structure or record, and relationships can be defined between these structures and records.

When working with data files, an application must be coded to work with the specific structure of each data file. In contrast, a database contains a catalog that applications use to determine how data is organized. Generic database applications can use the catalog to present users with data from different databases dynamically, without being tied to a specific data format.

Although there are different ways to organize data in a database, relational databases are one of the most effective. Relational database systems are an application of mathematical set theory to the problem of effectively organizing data.

In a relational database, data is collected into tables (called relations in relational theory).

When organizing data into tables, you can usually find many different ways to define tables. Relational database theory defines a process called normalization, which ensures that the set of tables you define will organize your data effectively.

SQL Server 2000 supports having a wide range of users access it at the same time. An instance of SQL Server 2000 includes the files that make up a set of databases and a copy of the DBMS software. Applications running on separate computers use a SQL Server 2000 communications component to transmit commands over a network to the SQL Server 2000 instance.

CHAPTER 3

SYSTEM ANALYSIS

System Analysis is the process of understanding a problem domain and the user requirements for the purpose of developing a computer application system to serve the needs of the users.

3.1 PROBLEM STATEMENT

The main objective of this system is to control all the activities of the organization through browser. This system should be user-friendly and should be able to provide access to the user who has appropriate username and password.

3.2 EXISTING SYSTEM

The existing system is the current status of the system and its current working. Server management, is normally done at the server side. So for any of the operations associated with the server, the physical presence of the human is a must, at the server.

Another additional task will be with databases, if the site involves database manipulations. Once the table is created within a database, the records may be forming along with the normal operations within the web pages. But, for modifying the structure of the tables or adding the new table itself is possible only if one goes to the backend itself. This too must be done at the server side only.

One important issue with any website is the access privileges that it provides to its users. There would be several registered users with any sites. There will be server who tries to hack or enter the site without permission and access information from within. So security is a large question with any website.

Locking of site for any particular user or user groups is not possible in the existing system.

3.2.1 Drawbacks of the Existing System

- Time consuming.
- Updating and Retrieval tasks are very tedious.
- Highly error prone.
- Expensive: - The cost for purchasing different software is not affordable to small companies.
- Lack of Experienced Personnel: - Technically qualified persons are needed for doing the operations and managing the database.
- Poor Reliability: - Since the data is stored only in the company server, it is prone to data loss.

3.3 PROPOSED SYSTEM

The existing system is studied and some drawbacks were found. The proposed system involves a customized form of server management. This new application has brought an innovative feature in the current methods of web server management. This web-based application enables us to manage or administer a server, located at the remote place.

Here in this mode of server administration, there is no necessity that one has to be present physically at the server for performing various site management activities. There are lot of activities to be performed with a web site like modifying the site contents, adding new links and pages to the site etc.

The application allows one to do all these process from the client itself, with no necessity for present at the server itself.

Facilities also provide to alter or modify the tables present within the site with no fear of data loss. There is no need to go to backend at all, for these processes. We can perform all these application from the front end and that too from the client side itself. A GUI based interactive environment is provided to create and edit the

tables present in the site and also to modify the structure of the table, allowed by the database management system.

Another additional feature of this application is that it can be used with any site, no matter whether the site is a new one or a currently running one. The only necessity is that the site should be configured initially, so that the application can take full control, or all the modules can work effectively.

This project provides users to interact with the DBMS engine with the help of an elegant user interface. This application will provide facilities to create a new database, open an existing Database, export/import a database, delete a database, etc. Once a database is opened or created, facilities are provided for the creation of tables and views using a very user-friendly interface. This application also lists the entire tables created within a database, so that we may alter or drop a table, rename the object already created, display the records of a table, etc. This module also comes up with a query building option which will provide features of parsing SQL statements like insert, update, delete etc.

Other Features are :

- Simple and Easy to Use.
- File and Directory manipulations
- Grouping and authentication
- Focused for lower end DBMS users.
- Being compact can be associated with any other application.
- Multi Database Support.
- Facilities for Export and import of database.
- Compliant with ODBC Data Management Technique.

3.3.1 Advantages of the Proposed System

- By using this system, database can be managed through the Internet.
- The system can be customized for specific user needs.
- It is possible for the web developers to connect to the database to view the data or change the database design.
- Most of the data transformation services can be initiated through the web page template.
- This aims to eliminate the problems of current technologies by offering data and computational service inexpensively to another company regardless of the hardware or software platform.
- Any user can manage their database without having a SQL Server client installed at their local machine.
- Reduction in cost.
- Provision for creating databases, tables, views, stored procedures, triggers and almost all the operations that can be done with a database.
- It is used to obscure data that is in plain view so that the normal user will not even be aware of extra messages

CHAPTER 4

SYSTEM DESIGN

System design is the most creative and challenging phase in the life cycle of system development. The first step to determine is what input data is needed to form the system and the database that has to be designed should meet the requirement of the proposed system. The next step is to determine how the output is produced and in what format.

4.1 ELEMENT OF DESIGN

The following are the elements of the system design;

- ❖ Input design
- ❖ Output design
- ❖ Architectural design
- ❖ Database design

4.1.1 Input design

The input design is the process of converting the user-oriented inputs into the computer-based format. The goal of designing input data is to make the automation as easy and free from errors as possible.

The input design requirement such as user friendliness, consistent format and interactive dialogues for giving the right message and help for the user at right time are also considered for the development of the project.

Throughout the system, the forms used to get input are,

- a. Admin login screen: The input fields accepted from the administrator are the user name and password (Appendix Figure A1.2)
- b. Group manager screen: This contains group id and password. Only the registered groups can use this form. (Appendix Figure A1.4)
- c. Information hiding screen: Using this form information is hidden and saved in the server. (Appendix Figure A 1.10)

- d. File/Directory screen: The file and directory operations such as file uploading and delete etc can be done using this form.(Appendix Figure A1.7 & 1.8)
- e. Database management screen: This screen is used to carry the database operation in a user-friendly way within the web page itself. (Appendix Figure A1.5 & 1.6)

4.1.2 Output design

The outputs are mainly used to communicate with the user, processing the input data given by the user. It is documented in each stage of the project to ensure error free output. The output screens are designed in a very simple and easy to understand format. All user options are presented in well-formatted forms. The user friendliness of the output screens makes it very easy to use. The quality refers to the way by which the output is presented to the user.

Main Outputs

- a. Group member view: This form deals with the group members in the group. Various outputs are generated within the system. The display provides a full-fledged view regarding the various users of the system.
- b. File created/deleted information: It allows the server manager to know of various status of the file system and also about the updating that are taken place within the server.
- c. Schedule screen: It allows the users to know about the schedules that are allocated by the user.

Outputs are also generated for various processes like table creations, and user groups etc of the site.

4.1.3 Architectural design

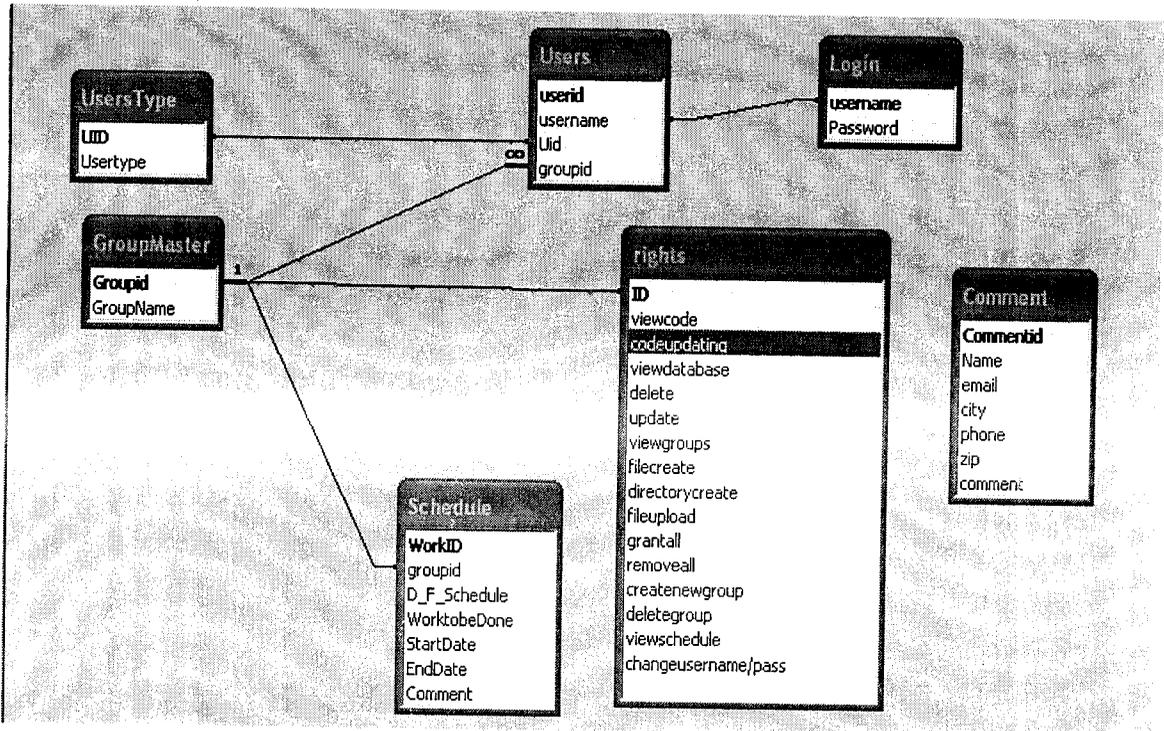


Figure 4.1 E-R Diagram

4.1.4 Database Design

A database is a collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective of database design is to make the data access easy, inexpensive and flexible to the user. The design of the database is one of the most critical parts of design phase. An elegantly database can play as a strong foundation for the whole system. The details about the data relevant for the system are identified first.

The data types for each data item in the tables are decided. For the optimum design of the database, to have better response time, to have data integrity, to

avoid redundancy and for the security of the database all the tables created are normalized. The database design is made up of two levels,

- ❖ Conceptual level
- ❖ Normalization

Normalization

The normalization simplifies the entities, removes the redundancies from the system data and finally builds a data structure, which is both flexible and adaptable to the system. Normalization offers a systematic step-by-step approach towards this goal. The different normal forms applied are given below,

- ❖ First normal form (1NF)
- ❖ Second normal form (2NF)
- ❖ Third normal form (3NF)

First Normal Form

A relation is said to be in 1NF if all the underlying domain of attributes contain simple individual values.

Second Normal Form

The 2NF is based on the concept of full function dependency. A relation said to be in 2NF if and only if it is in 1NF and every non-key attribute is fully functional dependent on candidate key of the table

Third Normal Form

The 3NF is based on the concept of transitive dependency. A relation in 2NF is said to be in 3NF if every non-key attribute is non-transitively

4.2 TABLE DESIGN

The table design containing Field Name, Data Type, Size of the field, Description and information about keys are represented. The finalized database table designs are as follows.

Table Name : Login

Description : This table holds login information by administrator, code manager and database manager

Field Name	Data Type	Field Size	Constraints	Description
Username	Varchar2	12	Primary Key	User Name
Password	Varchar2	10		Password

Table 4.1

Table Name : Group master

Description : This table holds the information about the group.

Field name	Data Type	Field Size	Constraints	Description
Groupid	Number	5	Primary Key	Group id
Group_name	Varchar2	25		Group name

Table 4.2

Table Name : Schedule

Description : This table holds the information about schedule .

Field Name	Data type	Field Size	Constraints	Description
Work_Id	Number	5	Primary Key	Work id
Groupid	Varchar2	10	Foreign Key	Id of group
D_F_Schedule	Date			Schedule date
Worktobedone	Varchar2	30		Work to be done
Startdate	Date			Starting date
Enddate	Date			Ending date
Comment	Varchar2	100		Comments

Table 4.3

Table Name : Comment

Description : This table is used to store contact information.

Field Name	Data Type	Field Size	Constraints	Description
Comment_id	Varchar2	8	Primary Key	Identification number
Name	Varchar2	19		Name
E_mail	Varchar2	20		Mail id
Phone	Number	15		Contact number
City	Varchar2	10		city
State	Varchar2	10		State
Zip	Number	8		Zip
Comment	Varchar2	50		Comment

Table 4.4

Table Name : Users type

Description : This table holds the user type information.

Field Name	Data Type	Field Size	Constraints	Description
Uid	Varchar2	5	Primary Key	User id
Users type	Varchar2	15		Category of user

Table 4.5

Table Name : Rights

Description : This table holds the information about the rights assigned to different groups.

Field Name	Data Type	Size	Constraints	Description
GroupId	Varchar2	5	Foreign Key	GroupId
Viewcode	Varchar2	10		To view code
Codeupdating	Number	15		To update code
View database	Varchar2	10		To view database
Delete	Varchar2	10		For deleting
Update	Number	8		To update
Viewgroups	Varchar2	10		To view the groups
Filecreate	Varchar2	12		To create the file
Dircreate	Varchar2	12		To create directory
Fileupload	Varchar2	12		For file uploading
Grantall	Varchar2	12		To grant all rights
Removeall	Varchar2	10		To revoke all rights
Createnewgroup	Varchar2	10		To create new group
Deletegroup	Varchar2	10		To delete the group
Viewscehdule	Varchar2	10		To view the schedule
Change username/pass	Varchar2	10		Change username/pass

Table 4.6

Table name : users

Description : This table is used to hold the user information

Field Name	Data Type	Field Size	Constraints	Description
userid	Varchar2	12	Primary Key	userid
Username	Varchar2	10		User name
Uid	Varchar2	5	Foreign Key	Category of user
groupid	Number	5		Group id

Table 4.7

CHAPTER 5

DATA FLOW DIAGRAM

5.1 Data Flow Diagram (DFD)

The data flow diagram is graphical representation which depicts the information regarding the flow of control and the transformation of data from input to output. The dataflow may be used to represent the system or software at any level of abstraction. In fact dataflow diagram may be partitioned into levels. A level 0 data flow diagram is called the Context Diagram, which represents the entire software element as single bubble with input and output arrows.

DFD Level 0:

This diagram shows the level 0 data flow diagram which represents the entire software.

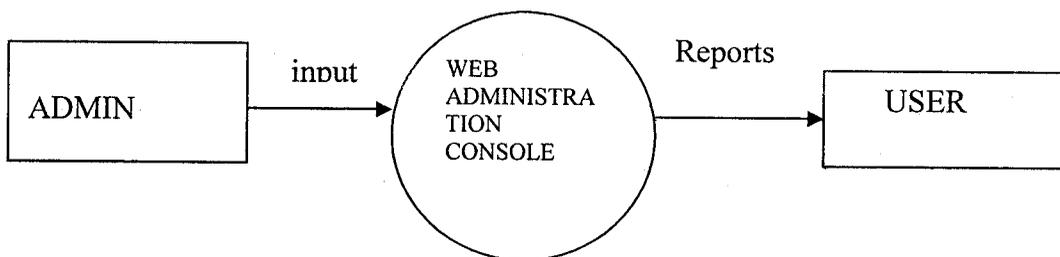


Figure 5.1

DFD Level 1:

This diagram represents the data flow diagram level 1 .

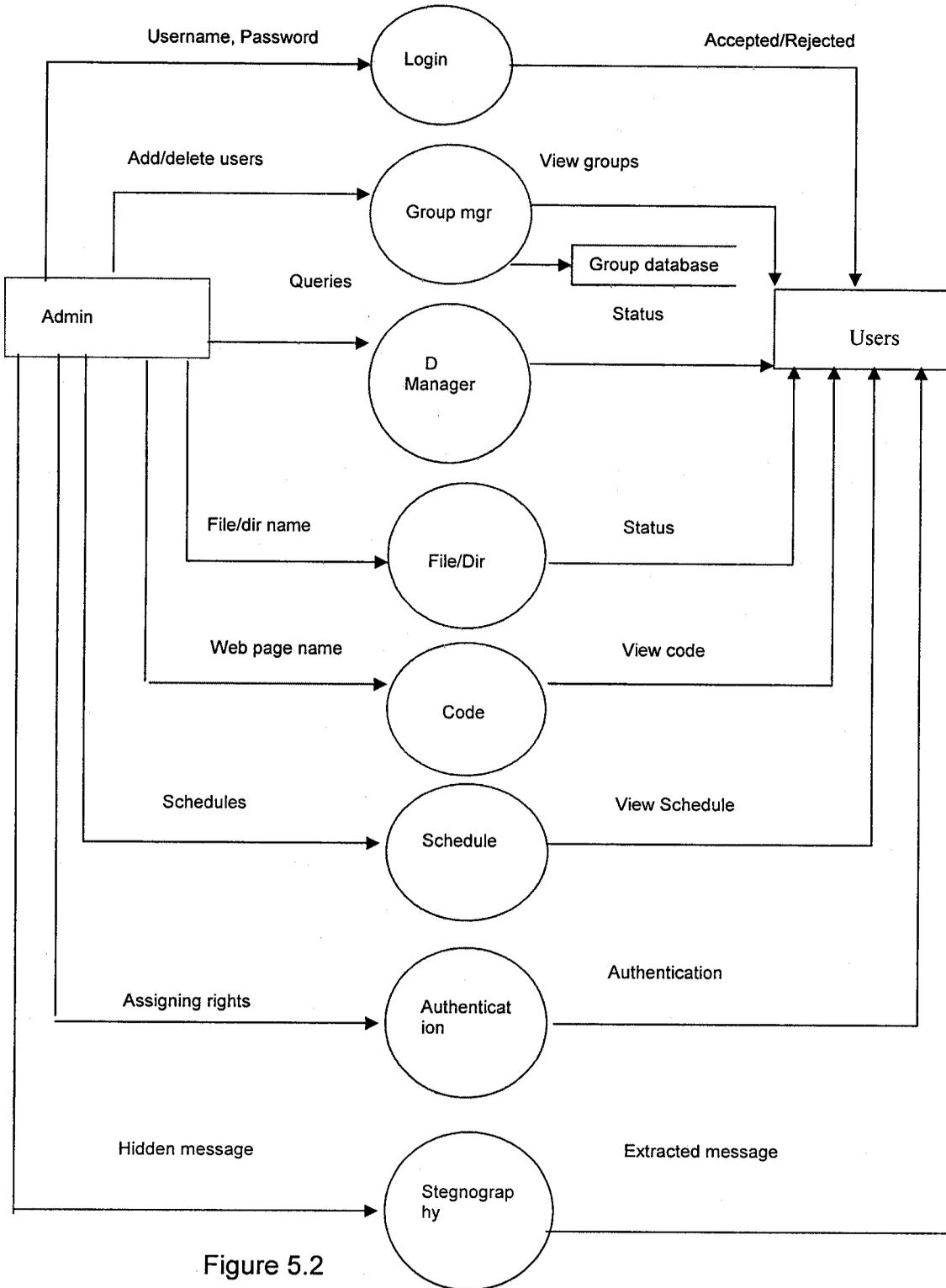


Figure 5.2

DFD Level 2:

The data flow diagram level 2 describes the function of each user type.

DFD Level 2.1 The administrator performs various operations.

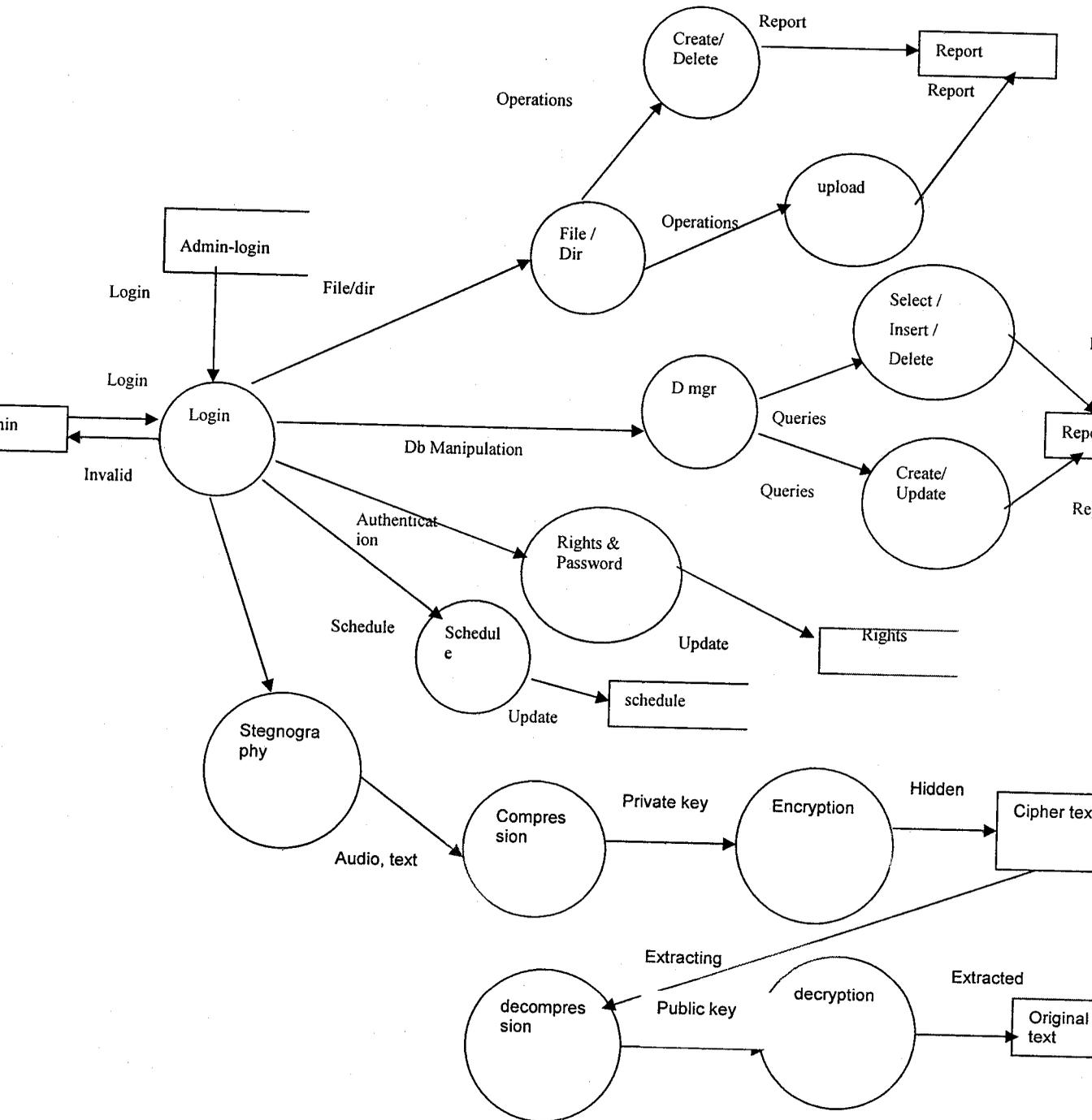


Figure 5.3 DFD FOR ADMINISTRATOR

DFD Level 2.2

This module accepts login and password and validates authorization using Login table. If it is a valid user, then the module guides them further to add, delete and view the existing users.

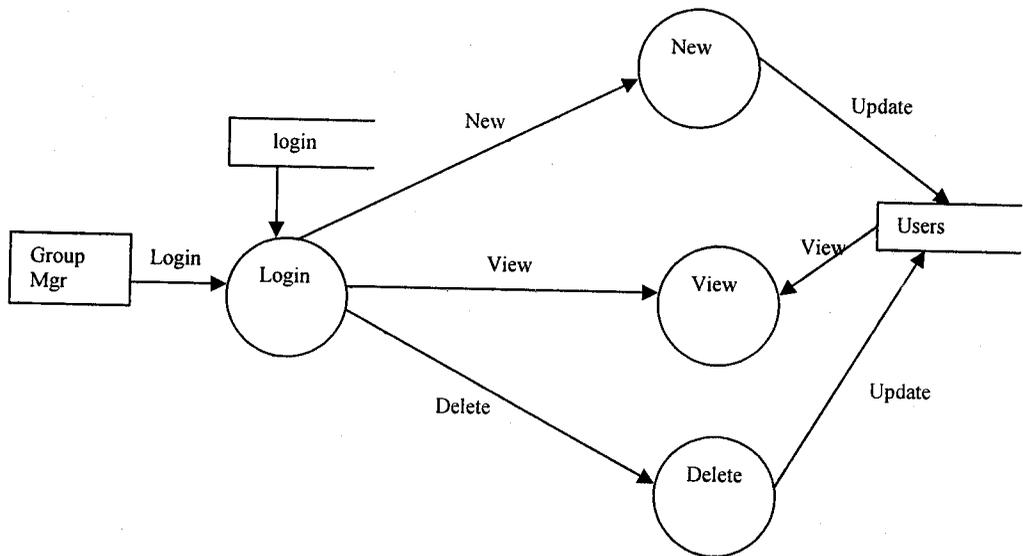


Figure 5.4 DFD FOR GROUP MANAGER

DFD Level 2.3

This module accepts login and password and validates authorization using Login table. If it is a valid user, then the module guides them further to save and view the web page.

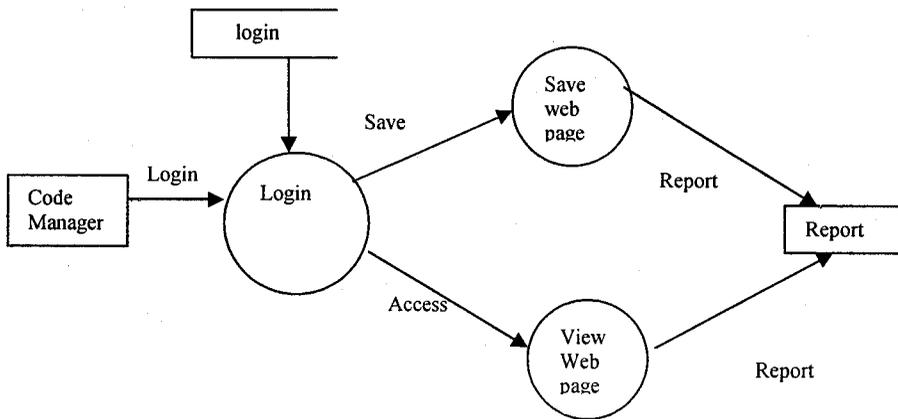


Figure 5.5 DFD FOR CODE MANAGER



CHAPTER 6

SYSTEM TESTING AND IMPLEMENTATION

6.1 TEST PLAN

Testing is the process of executing a program with the indent of finding any errors. Testing is vital to the success of the system. Without proper testing, hidden errors will surface after some time of use and perhaps irreversible damage has been done to valuable data. A successful test is one that uncovers as yet undiscovered error. A series of tests like responsiveness, its value, stress and security are performed before the system is ready for user acceptance testing.

S NO	Test case	Expected Result	Actual Result
1	If user is an Admin	File and dir Manipulations	Refer Fig A 1.2
2	Upload File	Files Should be uploaded	Refer Fig A 1.7
3	Directory Creation, Upload and Deletion	Directories should be created, Uploaded and deleted	Refer Fig A 1.8 & A 1.9
4	If User is a group Manager	Adding and deleting users	Refer Fig A 1.4
5	New user Entry	Successful user Entry	Refer Fig A 1.4
6	Delete users	Successful user Deletion	Refer Fig A 1.3
7	If user is a D manager	Should allow to insert, delete and update queries	Refer Fig A 1.5

Table 6.1 Test Plan for Web-Admin

6.2 SYSTEM IMPLEMENTATION

Implementation is the stage where the theoretical designs are turned into working system. The most crucial stage in achieving success of the new system is in giving confidence to the users that will work efficiently and effectively.

The primary goal of implementation is to write source code to its specification which can be easily verified, and so that debugging, testing, and modification can be eased. The goal can be achieved by making the source code as clear and straight forward as possible. The implementation is the process of converting a new or revised system into operational one. It is the key stage in achieving a new system because it involves a lot of upheaval in the user environment.

During the course of the project various risks like computer failure leading to delay and data loss were identified. Timely measures were taken to minimize them. Similarly, risk of data loss was minimized by taking regular backups. Thus, by eliminating the risk of delay and data loss, the schedule was kept on track. By keeping to schedule, the risks of missing deadlines were reduced.

6.2.1 Algorithm implemented for Stegnography

Huffman compression algorithm is named after its inventor, David Huffman, formerly a professor at MIT .Huffman compression is a lossless compression algorithm that is ideal for compressing text or program files.

How Huffman Compression works :

Huffman compression belongs into a family of algorithms with a variable codeword length. That means that individual symbols (characters in a text file for instance) are replaced by bit sequences that have a distinct length. So symbols that occur a lot in a file are given a short sequence while other that are used seldom get a longer bit sequence.

Example :

The string "go go gophers" is encoded in ASCII and save bits using a simple coding scheme, and Huffman coding is used to compress the data in which more bits are saved.

With an ASCII encoding (8 bits per character) the 13 character string "go go gophers" requires 104 bits. The table below on the left shows how the coding works.

coding a message

ASCII coding			3-bit coding		
char	ASCII	binary	char	code	binary
g	103	1100111	g	0	000
o	111	1101111	o	1	001
p	112	1110000	p	2	010
h	104	1101000	h	3	011
e	101	1100101	e	4	100
r	114	1110010	r	5	101
s	115	1110011	s	6	110
space	32	1000000	space	7	111

The string "go go gophers" would be written (coded numerically) as 103 111 32 103 111 32 103 111 112 104 101 114 115. Although not easily readable by humans, this would be written as the following stream of bits (the spaces would not be written, just the 0's and 1's)

```
1100111 1101111 1100000 1100111 1101111 1000000 1100111 1101111
1110000 1101000 1100101 1110010 1110011
```

Since there are only eight different characters in "go go gophers", it's possible to use only 3 bits to encode the different characters. For example, use the encoding in the table on the right above, though other 3-bit encodings are possible.

Now the string "go go gophers" would be encoded as 0 1 7 0 1 7 0 1 2 3 4 5 6 or, as bits:

000 001 111 000 001 111 000 001 010 011 100 101 110 111

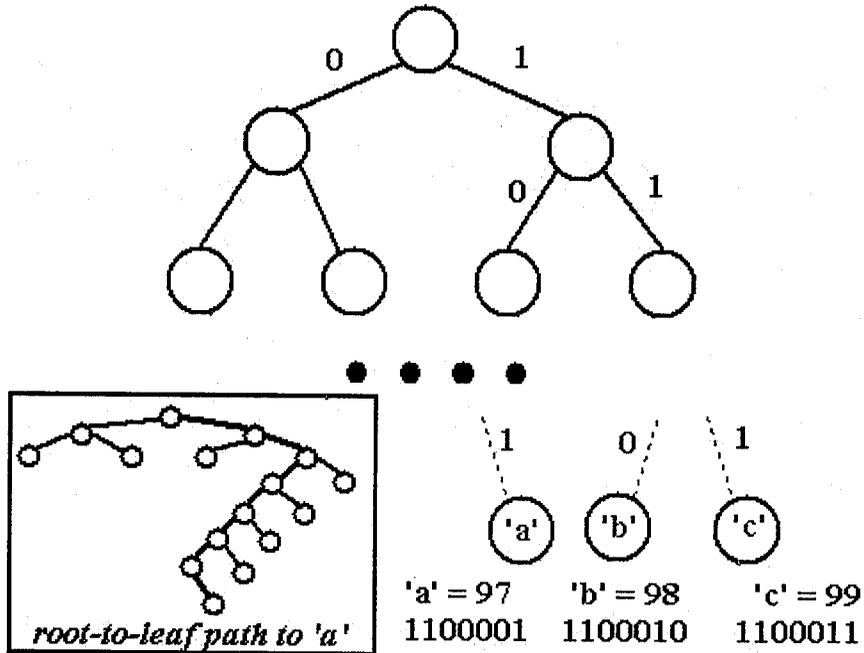
By using three bits per character, the string "go go gophers" uses a total of 39 bits instead of 104 bits. More bits can be saved if fewer than three bits to encode characters like g, o, and space that occur frequently and more than three bits to encode characters like e, p, h, r, and s that occur less frequently in "go go gophers". This is the basic idea behind Huffman coding: to use fewer bits for more frequently occurring characters. This is done using a tree that stores characters at the leaves, and whose root-to-leaf paths provide the bit sequence used to encode the characters.

Towards a Coding Tree

A tree view of the ASCII character set

Using a tree (actually a binary trie, more on that later) all characters are stored at the leaves of a complete tree. In the diagram to the right, the tree has eight levels meaning that the root-to-leaf path always has seven edges. A left-edge (black in the diagram) is numbered 0, a right-edge (blue in the diagram) is numbered 1. The ASCII code for any character/leaf is obtained by following the root-to-leaf path and catening the 0's and 1's. For example, the character 'a', which has ASCII value 97 (1100001 in binary), is shown with root-to-leaf path of

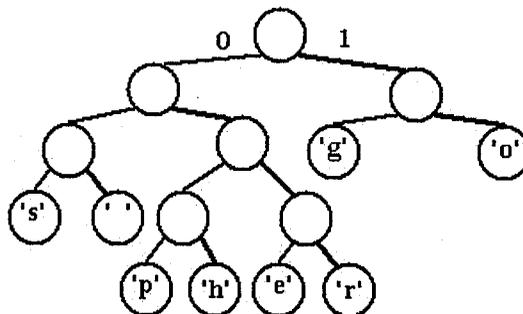
right-right-left-left-left-left-right.



The structure of the tree can be used to determine the coding of any leaf by using the 0/1 edge convention described. If different tree is used, different coding is obtained. As an example, the tree below on the right yields the coding shown on the left.

char binary

- 'g' 10
- 'o' 11
- 'p' 0100
- 'h' 0101
- 'e' 0110
- 'r' 0111
- 's' 000
- ' ' 001



Using this coding, "go go gophers" is encoded (spaces wouldn't appear in the bitstream) as:

10 11 001 10 11 001 10 11 0100 0101 0110 0111 000

This is a total of 37 bits, which saves two bits from the

encoding in which each of the 8 characters has a 3-bit encoding that is shown above. The bits are saved by coding frequently occurring characters like 'g' and 'o' with fewer bits (here two bits) than characters that occur less frequently like 'p', 'h', 'e', and 'r'.

The character-encoding induced by the tree can be used to decode a stream of bits as well as encode a string into a stream of bits. You can try to decode the following bitstream; the answer with an explanation follows:

```
01010110011100100001000101011001110110001101101100000010
101 011001110110
```

To decode the stream, start at the root of the encoding tree, and follow a left-branch for a 0, a right branch for a 1. When leaf is reached, write the character stored at the leaf, and start again at the top of the tree. To start, the bits are 010101100111. This yields *left-right-left-right* to the letter 'h', followed (starting again at the root) with *left-right-right-left* to the letter 'e', followed by *left-right-right-right* to the letter 'r'. Continuing until all the bits are processed yields

her sphere goes here

Huffman Coding

Huffman's algorithm is used to construct a tree that is used for data compression. Assume that each character has an associated weight equal to the number of times the character occurs in a file, for example. In the "go go gophers" example, the characters 'g' and 'o' have weight 3, the space has weight 2, and the other characters have weight 1. When compressing a file calculation of these weights are needed, ignore this step for now and assume that all character weights have been calculated. Huffman's algorithm assumes that a single tree is built from a group (or forest) of trees. Initially, all the trees have a single node with a character and the character's weight. Trees are combined by picking two trees, and making a new tree from the

two trees. This decreases the number of trees by one at each step since two trees are combined into one tree. The algorithm is as follows:

Begin with a forest of trees. All trees are one node, with the weight of the tree equal to the weight of the character in the node. Characters that occur most frequently have the highest weights. Characters that occur least frequently have the smallest weights.

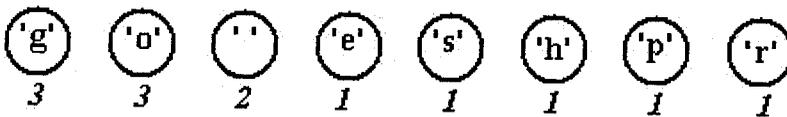
Repeat this step until there is only one tree:

Choose two trees with the smallest weights, call these trees T_1 and T_2 .

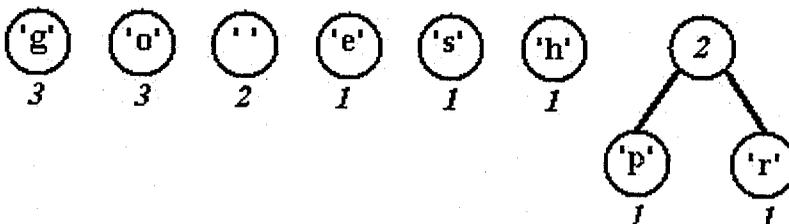
Create a new tree whose root has a weight equal to the sum of the weights $T_1 + T_2$ and whose left subtree is T_1 and whose right subtree is T_2 .

The single tree left after the previous step is an optimal encoding tree.

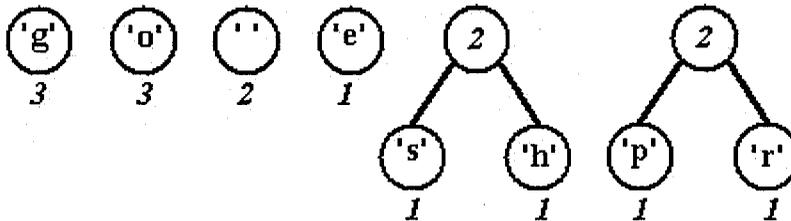
The string "go go gophers" is used as an example. Initially the forest is shown below. The nodes are shown with a weight/count that represents the number of times the node's character occurs.



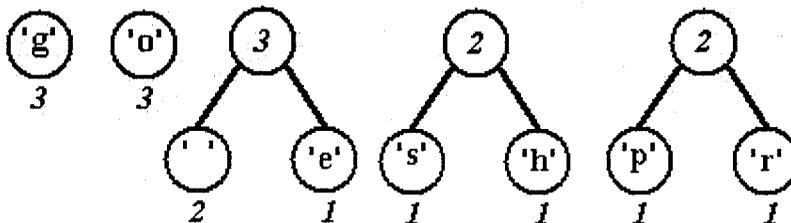
Pick two minimal nodes. There are five nodes with the minimal weight of one, it doesn't matter which two to pick. In a program, the deterministic aspects of the program will dictate which two are chosen, e.g., the first two in an array, or the elements returned by a priority queue implementation. A new tree is created whose root is weighted by the sum of the weights chosen. Now there are forest of seven trees as shown here:



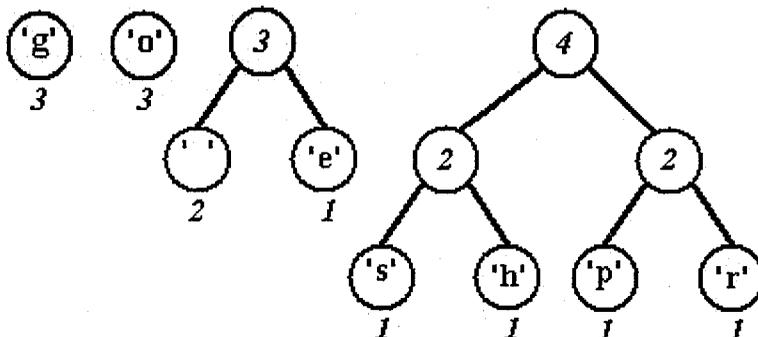
Choosing two minimal trees yields another tree with weight two as shown below. There are now six trees in the forest of trees that will eventually build an encoding tree.



Again choose the two trees of minimal weight. The lowest weight is the 'e'-node/tree with weight equal to one. There are three trees with weight two, choose any of these to create a new tree whose weight will be three

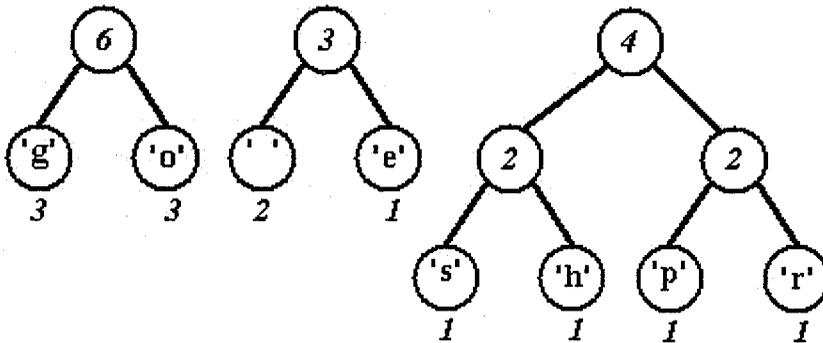


Now there are two trees with weight equal to two. These are joined into a new tree whose weight is four. There are four trees left, one whose weight is four and three with a weight of three.

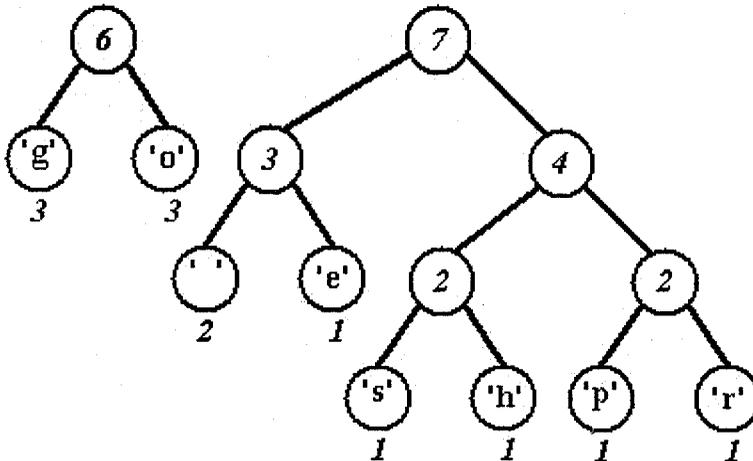


Two minimal (three weight) trees are joined into a tree whose weight is six. In the diagram below choose the 'g' and 'o' trees (the 'g' tree and the space-'e' tree or the 'o' tree and the space-'e' tree can be chosen.) There are three trees left.

Two minimal (three weight) trees are joined into a tree whose weight is six. In the diagram below choose the 'g' and 'o' trees (the 'g' tree and the space-'e' tree or the 'o' tree and the space-'e' tree chosen.) There are three trees left.

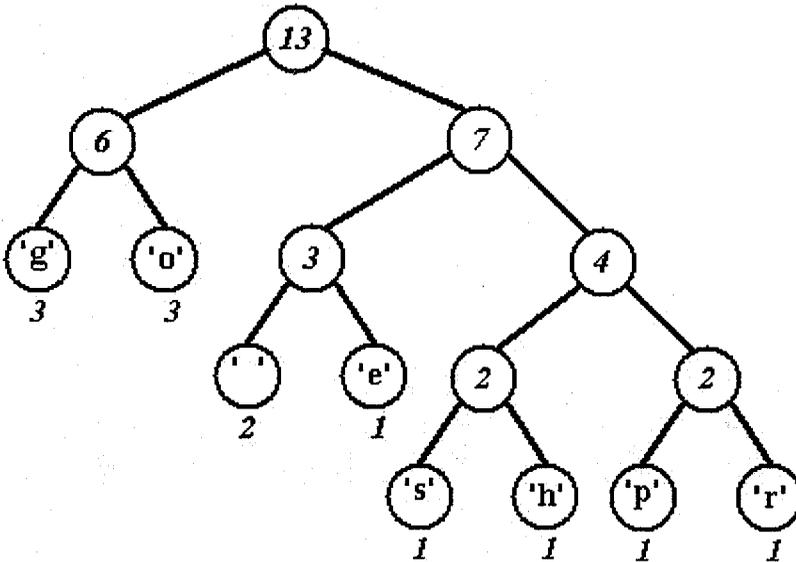


The minimal trees have weights of three and four, these are joined into a tree whose weight is seven leaving two trees.



Finally, the last two trees are joined into a final tree whose weight is thirteen, the sum of the two weights six and seven. Note that this tree is

different from the tree we used to illustrate Huffman coding above, and the bit patterns for each character are different, but the total number of bits used to encode "go go gophers" is the same



The character encoding induced by the last tree is shown below where again, 0 is used for left edges and 1 for right edges

char	binary
'g'	00
'o'	01
'p'	1110
'h'	1101
'e'	101
'r'	1111
's'	1100
' '	100

The string "go go gophers" would be encoded as shown (with spaces used for easier reading, the spaces wouldn't appear in the real encoding).

00 01 100 00 01 100 00 01 1110 1101 101 1111 1100

Once again, 37 bits are used to encode "go go gophers". There are several trees that yield an optimal 37-bit encoding of "go go gophers". The tree that actually results from a programmed implementation of Huffman's algorithm will be the same each time the program is run for the same weights (assuming no randomness is used in creating the tree).

CHAPTER 7

CONCLUSION AND FUTURE ENHANCEMENT

7.1 CONCLUSION

The project entitled “Web Administration Console” is a web data administrator utility application. The Web Administration Console is very effective in controlling a site that involves complex database manipulations and helps to configure and co-ordinate various operations. Efficient administration is a vital part in the survival of any web site. Usually this is possible and is done at server side only.

The software is developed with scalability in mind. Additional modules can be added easily when needed. The software is developed with a modular approach. All modules in this system have been tested separately and put together to form the main system. Finally the system is tested with real data and all modules of the project running successfully. Thus the system has fulfilled all the objectives identified and is able to replace the existing system.

The software was tested and implemented with real data and were found to be error free. Also, it is found that the system will work successfully. The system is maximum user friendly. Adequate exception handling is done in all required parts of the project. Error messages will be provided whenever the user makes an error so that it is easy for the user to understand. Also security is the main consideration in this project. The system is protected from any unauthorized access by giving user name and password during log in process.

The project “Web Administration Console” is very useful because the site owners can manage their database available at the web server through the web application itself. All groups and users can manage their database available at the web server by using this project according to their rights provided by the administrator.

7.2 FUTURE WORK :

The attraction of web services is simplicity, firewall neutrality and lack of dependency on the implementation technology at the service end. Web Services has huge potential for the deployment of Internet-based and service-oriented architectures. Web services have the potential to become the standard for how application development is done in the 21st century. Since “Web Administration Console” is a web based utility application, it has all these qualities.

The system has been designed in such a way that it can be modified with very little effort when such a need arises in the future. The system has been found to work efficiently and effectively. Due to its higher user friendliness, others may use these documents as a prototype for developing similar application.

In the future some more enhancements can be done to this project. It is possible for giving the provision to create roles such as private , public. Currently all the users are getting the role of public. It can be limited by creating roles. So each can perform the operations, which is assigned to them by means of role. It is also possible to maintain a log file for the users. The log file will contain the login name and password of the users. It will be stored at the time of login. It will be useful for the administrator to know which users are entered and what they are doing. In this way the project “Web Administration Console” will give more security for the operations.

APPENDICES

Welcome Page

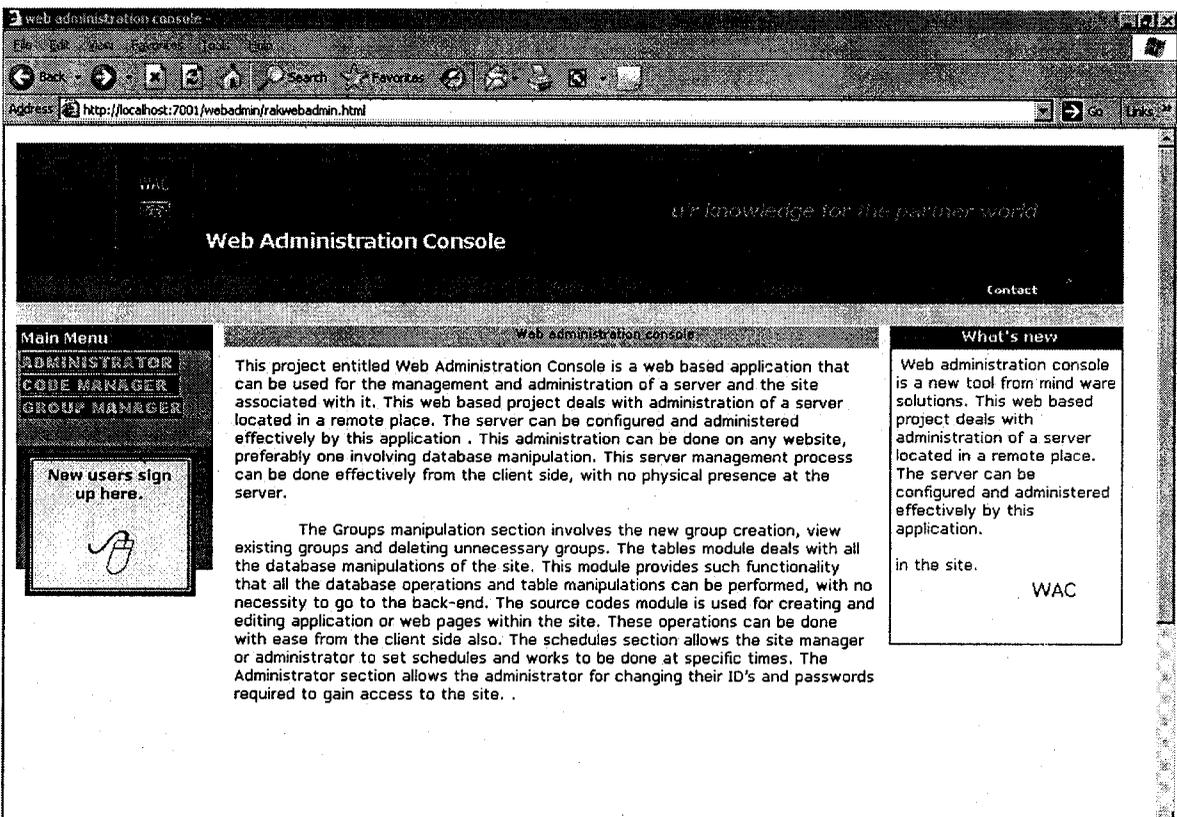


Figure A1.1 Welcome Page

Login Page

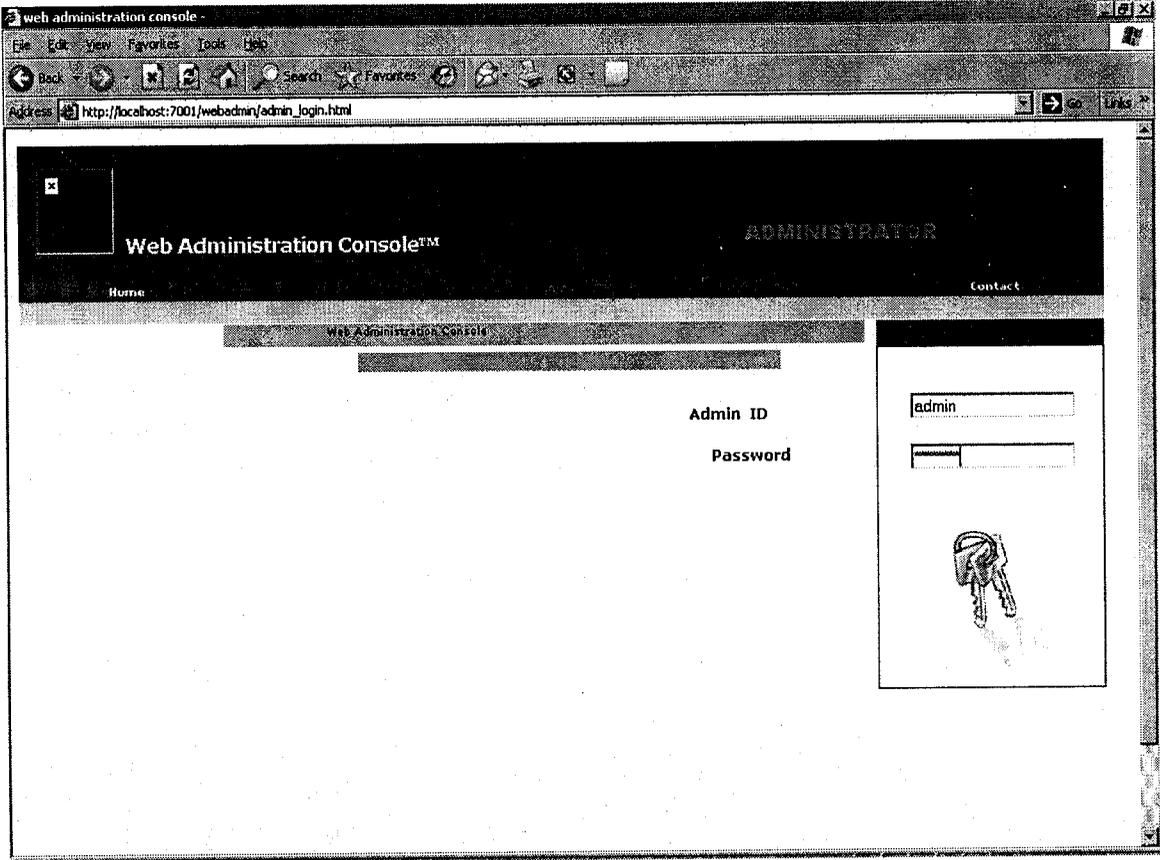


Figure A1.2 Admin Login Page

Assigning Rights

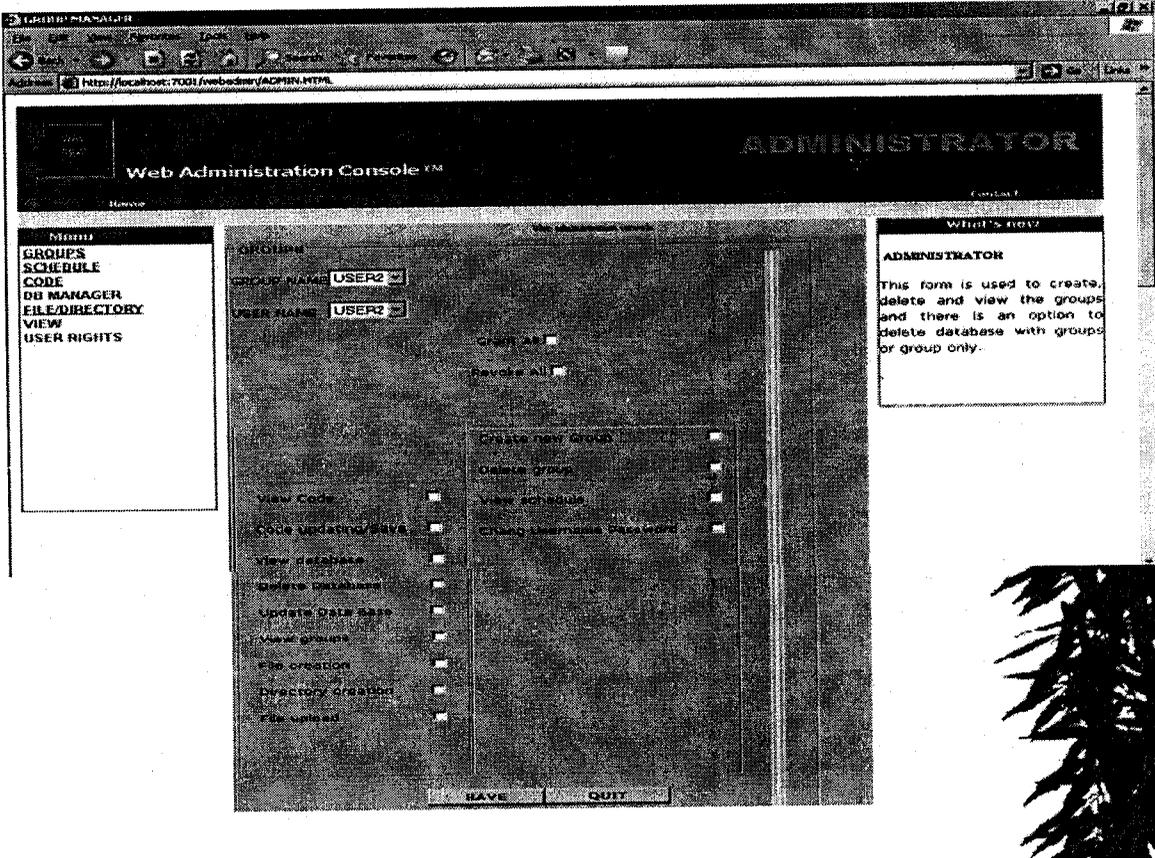


Figure A1.3 Assigning Rights

Group Manager

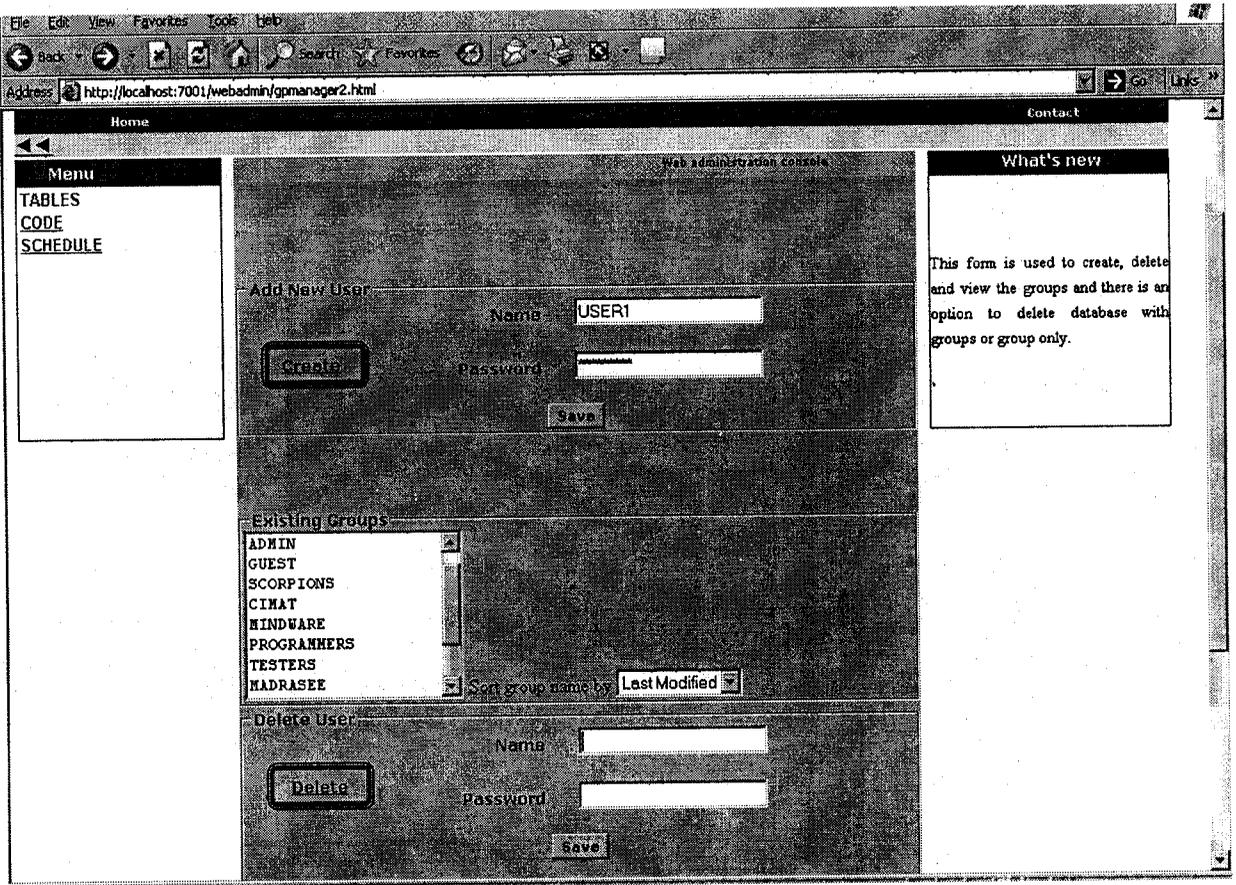


Figure A1.4 Adding New User

Database Manager-Query Selection

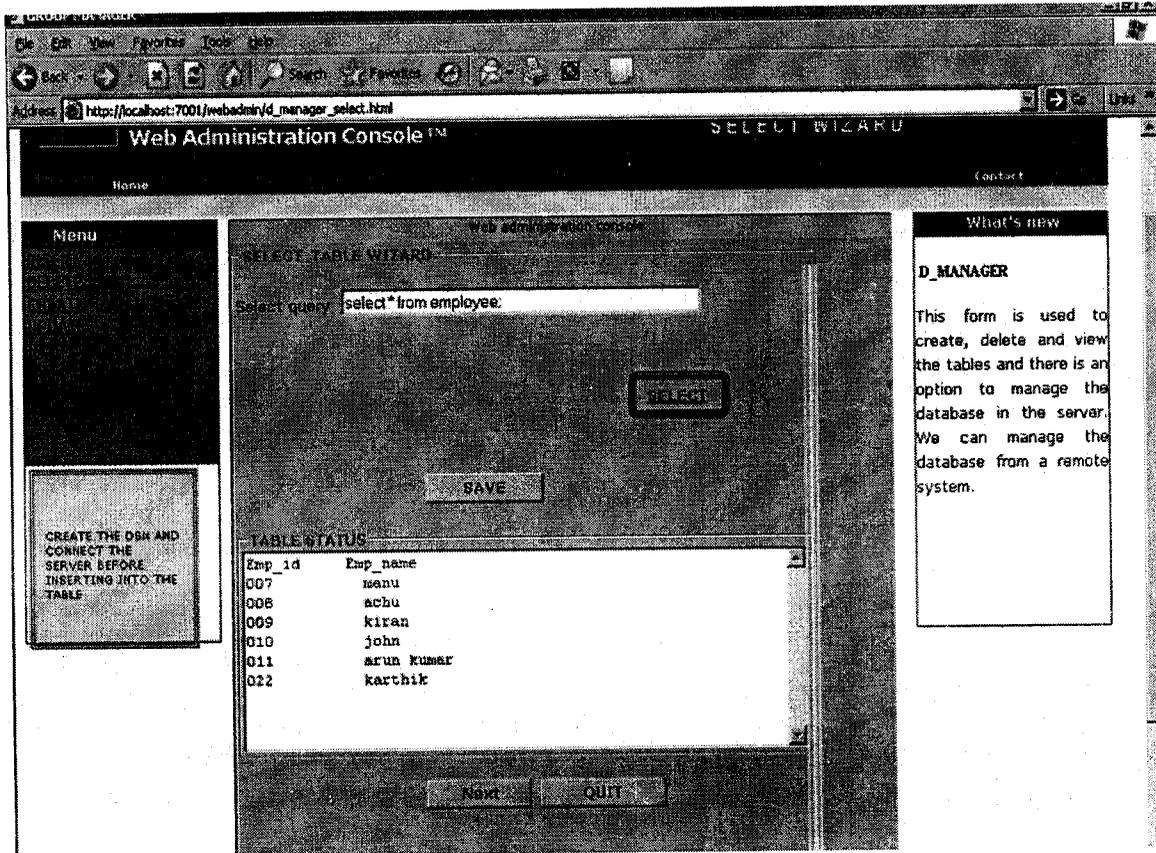


Fig A1.5 Select Query

Database Manager-Query Deletion

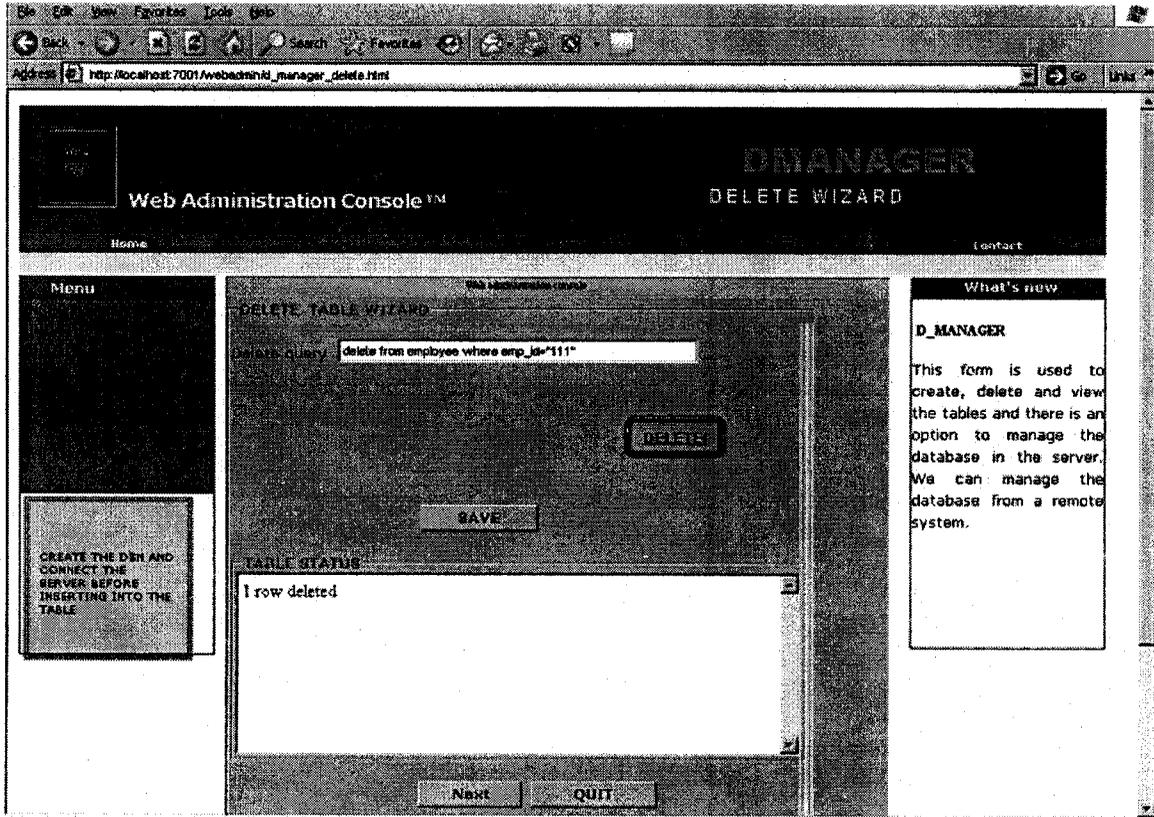


Fig A1.6 Delete Query

File Deletion

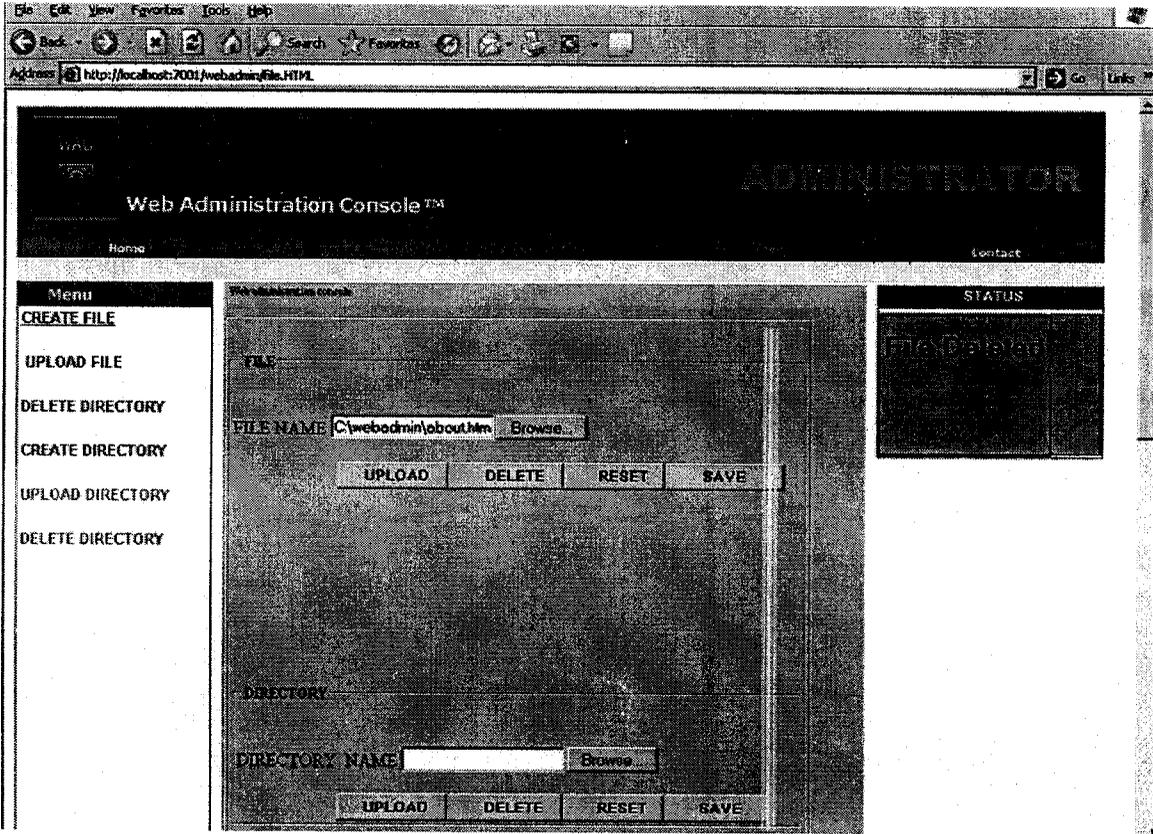


Figure A1.7 File Deletion

Directory Creation

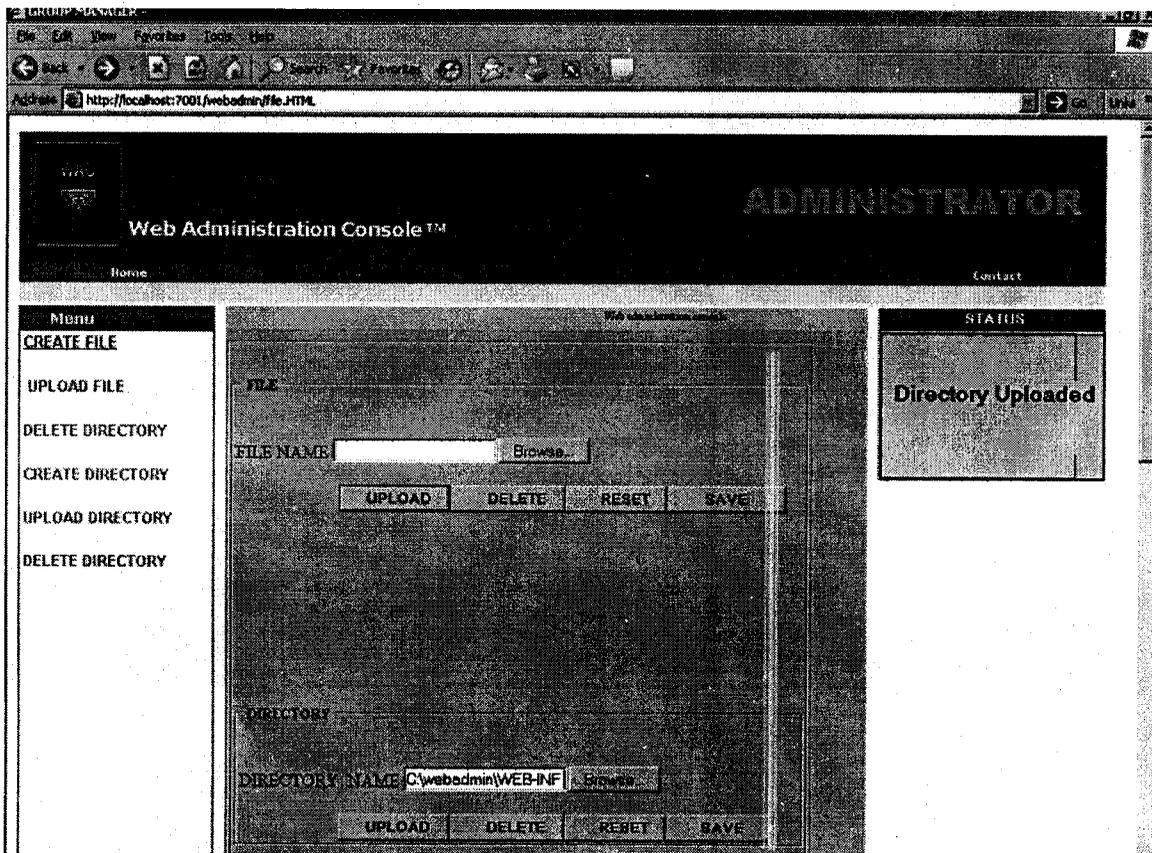


Figure A1.8 Directory Creation Page

Directory Deletion

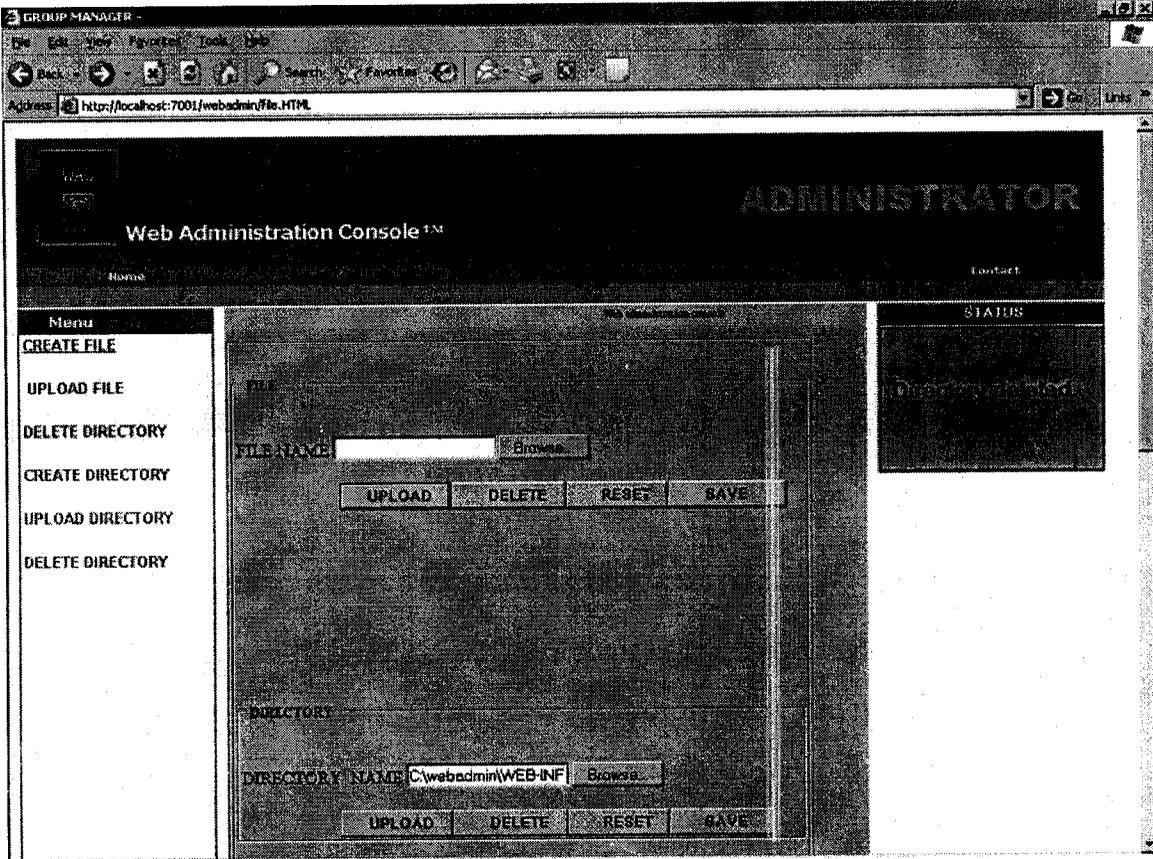


Figure A1.9 Directory Deletion Page

Steganography _ information hiding

Steganography For Wave

Settings

Wave File Browse...

Record a Wave on the fly

Key File Browse...

Encode

Save Result as: Browse...

Hide Content of File Browse...

Hide Text

Figure A1.10 information hiding

Steganography _ information retrieving

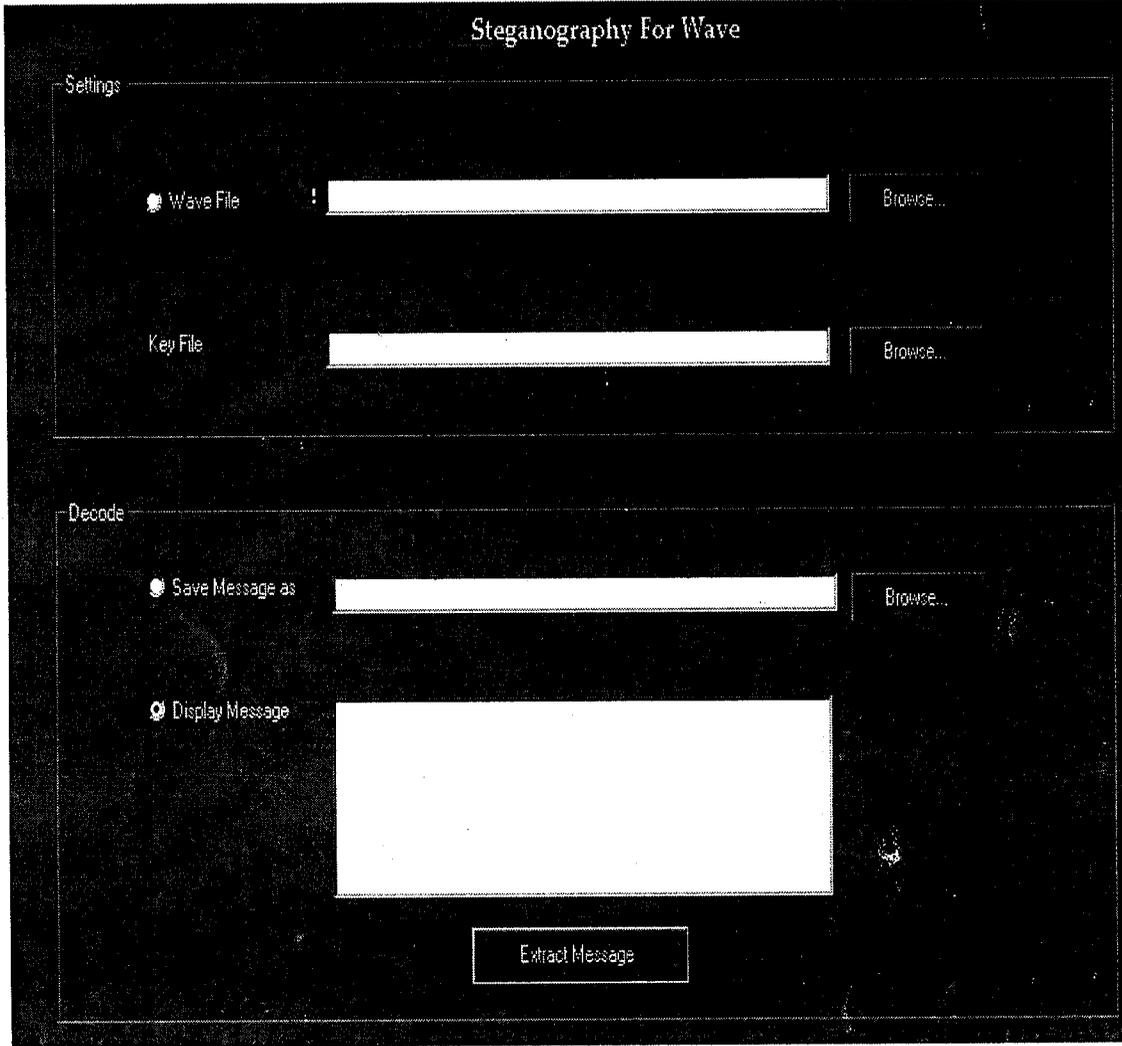


Figure A1.11 information hiding

REFERENCES

Books

1. Programming ADO.NET, January 2004, Richard Hundhausen, Steven Borg.
2. Visual Basic .NET Programming Black Book , Steven Holzner
3. VB.NET Programming Bible , Bill Evjen, Jason , Beres Et Al
4. Database System Concepts, Abraham Silberschatz, Henry, Sudarshan.
5. Software Engineering. A Practitioners Approach, Roger S. Pressman

Journals

1. Microsoft Developer Network Library 2003

Websites

<http://www.msdn.microsoft.com>

<http://www.webopedia.com>

<http://www.asp.net>

<http://www.programmersheaven.com>