

“PHYSICAL FILE STORAGE MANAGEMENT SYSTEM”

TABLE OF CONTENT

Acknowledgement	3
Declaration	4
1. Introduction and objectives	
1.1. Statement of Problem	8
1.2. Project Objectives	8
1.3. Project Category	9
1.4. Drawbacks of current Manual System	10
1.5. Proposed System	11
1.6. Needs	11
1.7. Software And Hardware Requirements	12
1.7.1. Introduction to PHP	14
1.8 Data Type	23
2. System Study	
2.1. Preliminary Investigation	27
2.2. System Development Life Cycle	28
2.3. Feasibility Study	32
2.4. Economic Feasibility	33
2.5. Technical Feasibility	36

2.6. Operational Feasibility	38
2.7. Feasibility Report	39
3. System Analysis	
3.1. Importance of Computerized System	41
3.2. Principles of System Analysis	42
3.3. System Analysis	43
3.4. Methods Used For Gathering Informations	46
3.5. Interviews and Questionnaires	48
4. System Design	
4.1. Design Objectives and Constraints	56
4.2. Physical Design	57
4.2.1. Design Methodology	57
4.2.2. Design Overview	57
4.2.3. Process modeling	58
4.3. Data Flow Diagram	58
4.4. Data Modeling	59
4.5. ER Diagram	63
4.6. Context Level DFD	64
4.7. Modules of the project	68
4.8. Process Logic	70
4.9. Scheduling	70
5. System Development	
5.1 Source Code	73
5.2 Code Efficiency	134

1. INTRODUCTION AND OBJECTIVE

1.1 STATEMENT OF PROBLEM

Physical File Storage Management System is an online web based system which is responsible for handling the management of Customers's file. In Physical File Storage Management System a customer and an employee of staff is being created by the administration. The Employee can be register itself by new employee signup form. All the files are created by employee after his login and File status is being issued to employee by the administrator of the company and after work of file employee can update the status of file is completed. The Employee can check his file issued history and can view his profile also. When file work will be completed and employee will update status of file by login.

Physical File Storage Management System is primarily meant for giving its services to its customers it gives full freedom to check the track of his files and check total files in storage. In PFSMS a customer can be created by administrator and a customer can be register himself by customer sign up form. Customers can also view their profile also. This project will automate all files status i.e. which file is created by which employee and for which customer, what is status (assigned or completed) and File detail (file name, file type, file location, file date, status).

Physical File Storage Management System , administrator can check the customers report, employees report, new created files details, issued to employee file reports, Beside that administrator can issue file to any employee and it will update the file status is Assigned.

1.2 PROJECT OBJECTIVE

The main objective of the Physical File Storage Management System is discipline of the planning, organizing and managing the hard copy file record storage tasks. Our project aims at making the task of file storage easy. Physical File Storage Management System is entering the records of new customer tax return file, retrieving the details of customer audit file and track of customer confidential file. We can issue customer file to internal staff for do their work and maintain their records and can also check which file issued to whom. In the project we can maintain the track report of customer confidential file, other files and Physical File

Storage Management system also send E-mail to management when customer files is issued to staff. The staff can search the customer files and management can also check the track report of customer confidential files. Physical File Storage Management System can generate historical data.

1.3 PROJECT CATEGORY

We have several alternative suggestions about the project category like: desktop application, web-based application, OOPs application, Networking, RDBMS etc. Out of available one we have opted Web based application based on client server architecture. The proposed system falls into the category of Multimedia. A web based system has two types of pages one is static web pages and another are dynamic web pages, which are saved on web server and can be seen by sending request to web server through HTTP protocol.

- **Static web pages:** - Static web pages are easy to spot sometimes we can pick them out by just looking at the content of the page. The content (text, images, hyperlinks, and so on) and appearance of static web pages is always the same regardless of who visits the page, or how and when they arrive at the page, or any other factors.
- **Dynamic web pages:** - The dynamic web pages are web pages, which we can interact like a time is shown on the page or we are sending information through web form, online chat, sending mail on net etc all the action this action performed with the help of dynamic web pages.
- **Web Server:** - Web server are software that manage web pages and make them available to client browser via local network or over the internet. In the case of the Internet, the web server and browser are usually on two different machines, possibly many miles apart. However, in a local situation you can set up machine that runs the web server software, and then use a browser on the same machine to look at its web pages.

1.4 DRAWBACKS OF CURRENT MANUAL- SYSTEM

- The current manual system has a lot of paper work and it does not deal with exact details.
- To maintain the records of sale and service manually, is a Time-consuming job.
- With the increase in database, it will become a massive job to maintain the database.
- Requires large quantities of file cabinets, which are huge and require quite a bit of space in the office, which can be used for storing records of previous claims.
- The retrieval of records of the employees,customers,files,their assignments will be a tedious job.
- Lack of security for the records, anyone disarrange the records of your system.

Establish The Need Of New System

1. **Problem of Reliability:** Current system is not reliable. It seems to vary in quality from one month to the next. Some times it gives good output, but some times the output is worst.
2. **Problem of Accuracy:** There are too many mistakes in reports.
3. **Problem of timeliness:** In the current system the reports and output produced is mostly late and in most of the cases it is useless because it is not on time.
4. **Problem of Validity:** The output and reports mostly contains misleading information. The customer's information is sometimes not valid.
5. **Problem of Economy:** The current system is very costly. We have to spend lots of money to keep the system up and going, but still not get the desired results.

6. **Problem of Capacity:** The current system is suffering from problem of capacity also. The staff for organization is very less and the workload is too much. Few peoples cannot handle all the work.

Some of the problems being faced in manual system are as follows:

- No control of customer confidential file
- Fast report generation is not possible.
- No Tracing a confidential or other files.
- Customer confidential or other files are misplace.
- Information about issue or return of the customer files are not properly maintained.
- No central database can be created as information is not available in database

1.5 PROPOSED SYSTEM

The Physical File Storage Management system is designed and developed for issue of confidential or others files in the storage room. The files received in the storage room are entered in files Entry form. When the staff member wants to get the specific file the same can be issued on the availability basis to the staff member. The issuance and issued date and returned date of the file is also entered.

1.6 NEED

I have designed the given proposed system in the JSP and oracle to automate the process. The following steps that give the detailed information of the need of proposed system are:

- **Performance:** During past several decades, the records are supposed to be manually handled for all activities. The manual handling of the record is time consuming and highly prone to error. To improve the performance of the Company's file system, the computerized system is to be undertaken. The computerized project is fully computerized and user friendly even that any of the members can see the report and status of the company's file system.
- **Efficiency:** The basic need of this website is efficiency. The website should be efficient so that whenever a new user submits his/her details the website is updated automatically. This record will be useful for other users instantly.

- **Control:** The complete control of the project is under the hands of authorized person who has the password to access this project and illegal access is not supposed to deal with. All the control is under the administrator and the other members have the rights to just see the records not to change any transaction or entry.
- **Security:** Security is the main criteria for the proposed system. Since illegal access may corrupt the database. So security has to be given in this project.

1.7 SOFTWARE & HARDWARE REQUIREMENTS

The need of today's software development is competence in a GUI based front-end tool, which can connect to Relational Database engines. This gives the programmer the opportunity to develop client server based commercial applications.

SOFTWARE SPECIFICATION

The software packages are used

- Operating System : Windows 7
- Front End : JSP, Java Script
- Back End : My SQL
- Server : Apache Tomcat Server 6.0
- Macromedia Dreamweaver 8
- Internet Explorer 6.0 or above

HARDWARE SPECIFICATON

On the basis of above software specification we have decided the following configuration of hardware for server and client.

- PROCESSOR – P4 3.06 GHz
- RAM- 256 MB
- HDD- 40 GB UPGRADABLE UPTO 80 GB
- CD-ROM
- MONITOR- VGA COLOR
- PRINTER- 132 COLUMN DOT MATRIX

FRONT END

JSP programming tools are complete programming environments. It allows programmers to build a GUI program using the various on-screen controls such as buttons, text, menus, boxes etc. These controls are placed on a form and then the processing details related with each control are filled in.

In the business world, competitive strategies have become the order of the day to improve quality, cut costs and provide a high response customer service base. Most organizations today need to be market driven and do a lot of value addition to their products and services. This naturally calls for rational decision making, which requires information. Information Technology or IT provides that effective channel to support and implement this strategy. Client/Server is the technology that empowers the desktop, thus setting a trend for the way successful organizations will use technology in the next decade.

Projecthelping.in

1.7.1 INTRODUCTION TO JSP

INTRODUCTION TO JSP

Adding dynamic content via expressions

As we saw in the previous section, any HTML file can be turned into a JSP file by changing its extension to .jsp. Of course, what makes JSP useful is the ability to embed Java. Put the following text in a file with .jsp extension (let us call it **hello.jsp**), place it in your JSP directory, and view it in a browser.

```
<HTML>
<BODY>
Hello! The time is now <%= new java.util.Date() %>
</BODY>
</HTML>
```

Notice that each time you reload the page in the browser, it comes up with the current time.

The character sequences `<%=` and `>` enclose Java expressions, which are evaluated at run time.

This is what makes it possible to use JSP to generate dynamic HTML pages that change in response to user actions or vary from user to user.

Exercise: Write a JSP to output the values returned by `System.getProperty` for various system properties such as `java.version`, `java.home`, `os.name`, `user.name`, `user.home`, `user.dir` etc.

Scriptlets

We have already seen how to embed Java expressions in JSP pages by putting them between the `<%=` and `>` character sequences.

But it is difficult to do much programming just by putting Java expressions inside HTML.

JSP also allows you to write blocks of Java code inside the JSP. You do this by placing your Java code between `<%` and `>` characters (just like expressions, but without the = sign at the start of the sequence.)

This block of code is known as a "scriptlet". By itself, a scriptlet doesn't contribute any HTML (though it can, as we will see down below.) A scriptlet contains Java code that is executed every time the JSP is invoked.

Here is a modified version of our JSP from previous section, adding in a scriptlet.

```
<HTML>
<BODY>
<%
    // This is a scriptlet. Notice that the "date"
    // variable we declare here is available in the
```

```

    // embedded expression later on.
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello! The time is now <%= date %>
</BODY>
</HTML>

```

If you run the above example, you will notice the output from the "System.out.println" on the server log. This is a convenient way to do simple debugging (some servers also have techniques of debugging the JSP in the IDE. See your server's documentation to see if it offers such a technique.)

By itself a scriptlet does not generate HTML. If a scriptlet wants to generate HTML, it can use a variable called "out". This variable does not need to be declared. It is already predefined for scriptlets, along with some other variables. The following example shows how the scriptlet can generate HTML output.

```

<HTML>
<BODY>
<%
    // This scriptlet declares and initializes "date"
    System.out.println( "Evaluating date now" );
    java.util.Date date = new java.util.Date();
%>
Hello! The time is now
<%
    // This scriptlet generates HTML output
    out.println( String.valueOf( date ));
%>
</BODY>
</HTML>

```

Here, instead of using an expression, we are generating the HTML directly by printing to the "out" variable. The "out" variable is of type [javax.servlet.jsp.JspWriter](#).

Another very useful pre-defined variable is "request". It is of type [javax.servlet.http.HttpServletRequest](#)

A "request" in server-side processing refers to the transaction between a browser and the server. When someone clicks or enters a URL, the browser sends a "request" to the server for that URL, and shows the data returned. As a part of this "request", various data is available, including the file the browser wants from the server, and if the request is coming from pressing a SUBMIT button, the information the user has entered in the form fields.

The JSP "request" variable is used to obtain information from the request as sent by the browser. For instance, you can find out the name of the client's host (if available, otherwise the IP address will be returned.) Let us modify the code as shown:

```

<HTML>
<BODY>
<%

```

```

// This scriptlet declares and initializes "date"
System.out.println( "Evaluating date now" );
java.util.Date date = new java.util.Date();
%>
Hello! The time is now
<%
    out.println( date );
    out.println( "<BR>Your machine's address is " );
    out.println( request.getRemoteHost());
%>
</BODY>
</HTML>

```

A similar variable is "response". This can be used to affect the response being sent to the browser. For instance, you can call `response.sendRedirect(anotherUrl);` to send a response to the browser that it should load a different URL. This response will actually go all the way to the browser. The browser will then send a different request, to "anotherUrl". This is a little different from some other JSP mechanisms we will come across, for including another page or forwarding the browser to another page.

Exercise: Write a JSP to output the entire line, "Hello! The time is now ..." but use a scriptlet for the complete string, including the HTML tags.

Mixing Scriptlets and HTML

We have already seen how to use the "out" variable to generate HTML output from within a scriptlet. For more complicated HTML, using the out variable all the time loses some of the advantages of JSP programming. It is simpler to mix scriptlets and HTML.

Suppose you have to generate a table in HTML. This is a common operation, and you may want to generate a table from a SQL table, or from the lines of a file. But to keep our example simple, we will generate a table containing the numbers from 1 to N. Not very useful, but it will show you the technique.

Here is the JSP fragment to do it:

```

<TABLE BORDER=2>
<%
    for ( int i = 0; i < n; i++ ) {
        %>
        <TR>
            <TD>Number</TD>
            <TD><%= i+1 %></TD>
        </TR>
    }
%>
</TABLE>

```

You would have to supply an int variable "n" before it will work, and then it will output a simple table with "n" rows.

The important things to notice are how the %> and <% characters appear in the middle of the "for" loop, to let you drop back into HTML and then to come back to the scriptlet.

The concepts are simple here -- as you can see, you can drop out of the scriptlets, write normal HTML, and get back into the scriptlet. Any control expressions such as a "while" or a "for" loop or an "if" expression will control the HTML also. If the HTML is inside a loop, it will be emitted once for each iteration of the loop.

Another example of mixing scriptlets and HTML is shown below -- here it is assumed that there is a boolean variable named "hello" available. If you set it to true, you will see one output, if you set it to false, you will see another output.

```
<%
  if ( hello ) {
    %>
    <P>Hello, world
    <%
  } else {
    %>
    <P>Goodbye, world
    <%
  }
%>
```

It is a little difficult to keep track of all open braces and scriptlet start and ends, but with a little practice and some good formatting discipline, you will acquire competence in doing it.

Exercise: Make the above examples work. Write a JSP to output all the values returned by System.getProperties with "
" embedded after each property name and value. Do not output the "
" using the "out" variable.

JSP Directives

We have been fully qualifying the java.util.Date in the examples in the previous sections. Perhaps you wondered why we don't just import java.util.*;

It is possible to use "import" statements in JSPs, but the syntax is a little different from normal Java. Try the following example:

```
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%
  System.out.println( "Evaluating date now" );
  Date date = new Date();
%>
Hello! The time is now <%= date %>
</BODY>
</HTML>
```

The first line in the above example is called a "directive". A JSP "directive" starts with <%@ characters.

This one is a "page directive". The page directive can contain the list of all imported packages. To import more than one item, separate the package names by commas, e.g.

```
<%@ page import="java.util.*,java.text.*" %>
```

There are a number of JSP directives, besides the page directive. Besides the page directives, the other most useful directives are include and taglib. We will be covering taglib separately.

The include directive is used to physically include the contents of another file. The included file can be HTML or JSP or anything else -- the result is as if the original JSP file actually contained the included text. To see this directive in action, create a new JSP

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<%@ include file="hello.jsp" %>
</BODY>
</HTML>
```

View this JSP in your browser, and you will see your original hello.jsp get included in the new JSP.

Exercise: Modify all your earlier exercises to import the java.util packages.

JSP Declarations

The JSP you write turns into a class definition. All the scriptlets you write are placed inside a single method of this class.

You can also add variable and method declarations to this class. You can then use these variables and methods from your scriptlets and expressions.

To add a declaration, you must use the `<%! and %>` sequences to enclose your declarations, as shown below.

```
<%@ page import="java.util.*" %>
<HTML>
<BODY>
<%!
    Date theDate = new Date();
    Date getDate()
    {
        System.out.println( "In getDate() method" );
        return theDate;
    }
%>
Hello! The time is now <%= getDate() %>
</BODY>
</HTML>
```

The example has been created a little contrived, to show variable and method declarations.

Here we are declaring a Date variable theDate, and the method getDate. Both of these are available now in our scriptlets and expressions.

But this example no longer works! The date will be the same, no matter how often you reload the page. This is because these are declarations, and will only be evaluated once when the page is loaded! (Just as if you were creating a class and had variable initialization declared in it.)

Exercise: Modify the above example to add another function `computeDate` which re-initializes `theDate`. Add a scriptlet that calls `computeDate` each time.

Note: Now that you know how to do this -- it is in general not a good idea to use variables as shown here. The JSP usually will run as multiple *threads* of one single instance. Different threads would interfere with variable access, because it will be the same variable for all of them. If you do have to use variables in JSP, you should use *synchronized* access, but that hurts the performance. In general, any data you need should go either in the *session* object or the *request* object (these are introduced a little later) if passing data between different JSP pages. Variables you declare inside *scriptlets* are fine, e.g. `<% int i = 45; %>` because these are declared inside the local scope and are not shared.

JSP Tags

Another important syntax element of JSP are tags. JSP tags do not use `<%`, but just the `<` character. A JSP tag is somewhat like an HTML tag. JSP tags can have a "start tag", a "tag body" and an "end tag". The start and end tag both use the tag name, enclosed in `<` and `>` characters. The end starts with a `/` character after the `<` character. The tag names have an embedded colon character `:` in them, the part before the colon describes the type of the tag.

For instance:

```
<some:tag>
body
</some:tag>
```

If the tag does not require a body, the start and end can be conveniently merged together, as

```
<some:tag/>
```

Here by closing the start tag with a `/>` instead of `>` character, we are ending the tag immediately, and without a body. (This syntax convention is the the same as XML.)

Tags can be of two types: loaded from an external tag library, or predefined tags. Predefined tags start with **jsp:** characters. For instance, `jsp:include` is a predefined tag that is used to include other pages.

We have already seen the `include` directive. `jsp:include` is similar. But instead of loading the text of the included file in the original file, it actually calls the included target at run-time (the way a browser would call the included target. In practice, this is actually a simulated request rather than a full round-trip between the browser and the server). Following is an example of `jsp:include` usage

```
<HTML>
<BODY>
Going to include hello.jsp...<BR>
<jsp:include page="hello.jsp"/>
</BODY>
</HTML>
```

Try it and see what you get. Now change the `"jsp:include"` to `"jsp:forward"` and see what is the difference. These two predefined tags are frequently very useful.

Exercise: Write a JSP to do either a forward or an include, depending upon a boolean variable (hint: The concepts of mixing HTML and scriptlets work with JSP tags also!)

JSP Sessions

On a typical web site, a visitor might visit several pages and perform several interactions.

If you are programming the site, it is very helpful to be able to associate some data with each visitor. For this purpose, "session"s can be used in JSP.

A session is an object associated with a visitor. Data can be put in the session and retrieved from it, much like a Hashtable. A different set of data is kept for each visitor to the site.

Here is a set of pages that put a user's name in the session, and display it elsewhere. Try out installing and using these.

First we have a form, let us call it GetName.html

```
<HTML>
<BODY>
<FORM METHOD=POST ACTION="SaveName.jsp">
What's your name? <INPUT TYPE=TEXT NAME=username SIZE=20>
<P><INPUT TYPE=SUBMIT>
</FORM>
</BODY>
</HTML>
```

The target of the form is "SaveName.jsp", which saves the user's name in the session. Note the variable "session". This is another variable that is normally made available in JSPs, just like out and request variables. (In the @page directive, you can indicate that you do not need sessions, in which case the "session" variable will not be made available.)

```
<%
String name = request.getParameter( "username" );
session.setAttribute( "theName", name );
%>
```

```
<HTML>
<BODY>
<A HREF="NextPage.jsp">Continue</A>
</BODY>
</HTML>
```

The SaveName.jsp saves the user's name in the session, and puts a link to another page, NextPage.jsp.

NextPage.jsp shows how to retrieve the saved name.

```
<HTML>
<BODY>
Hello, <%= session.getAttribute( "theName" ) %>
</BODY>
</HTML>
```

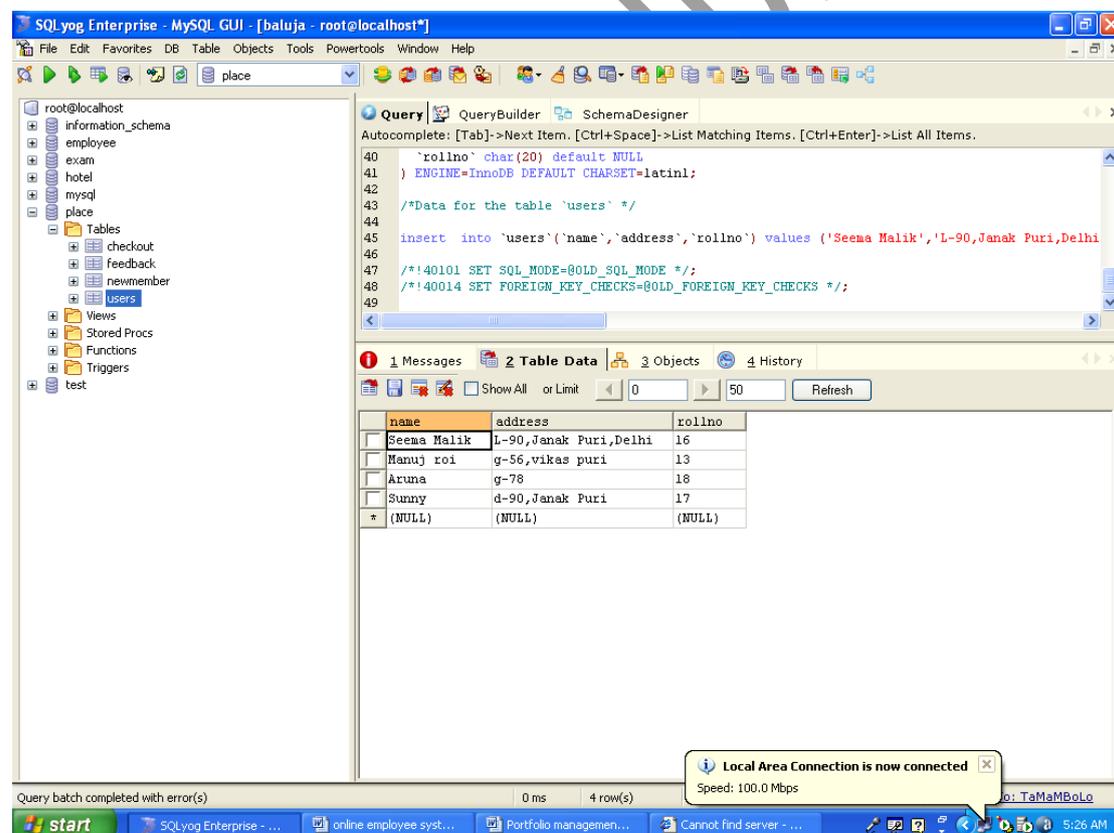
If you bring up two different browsers (not different windows of the same browser), or run two browsers from two different machines, you can put one name in one browser and another name in another browser, and both names will be kept track of.

The session is kept around until a timeout period. Then it is assumed the user is no longer visiting the site, and the session is discarded.

My SQL

Introduction

My SQL is an application used to create computer databases for the Microsoft Windows family of server operating systems. It provides an environment used to generate databases that can be accessed from workstations, the web, or other media such as a personal digital assistant (PDA). MY SQL is probably the most accessible and the most documented enterprise database environment right now. This also means that you can learn it a little quicker than most other database environments on the market



To start, you must have a computer that runs an appropriate operating system like Microsoft Windows >= XP Home Edition: that includes Windows XP Home Edition, Windows XP

Professional, Windows 2000 Professional, or any version of Windows Server 2003. In this case, you must install MY SQL Yog.

What is SQL Used for:

Using SQL one can create and maintain data manipulation objects such as table, views, sequence etc. These data manipulation objects will be created and stored on the server's hard disk drive, in a tablespace, to which the user has been assigned.

Once these data manipulation objects are created, they are used extensively in commercial applications.

DML, DCL, DDL:

In addition to the creation of data manipulation objects, the actual manipulation of data within these objects is done using SQL.

The SQL sentences that are used to create these objects are called DDL's or Data Definition Language. The SQL sentences used to manipulate data within these objects are called DML's or Data Manipulation Language. The SQL sentences, which are used to control the behavior of these objects, are called DCL's or Data Control Language.

1.8 DATA TYPE

. TEXT TYPES

CHAR()	A fixed section from 0 to 255 characters long.
VARCHAR()	A variable section from 0 to 255 characters long.
TINYTEXT	A string with a maximum length of 255 characters.
TEXT	A string with a maximum length of 65535 characters.
BLOB	A string with a maximum length of 65535 characters.
MEDIUMTEXT	A string with a maximum length of 16777215 characters.
MEDIUMBLOB	A string with a maximum length of 16777215 characters.
LONGTEXT	A string with a maximum length of 4294967295 characters.
LOB	A string with a maximum length of 4294967295 characters.

The () brackets allow you to enter a maximum number of characters will be used in the column.

VARCHAR(20)

CHAR and VARCHAR are the most widely used types. CHAR is a fixed length string and is mainly used when the data is not going to vary much in it's length. VARCHAR is a variable length string and is mainly used when the data may vary in length.

CHAR may be faster for the database to process considering the fields stay the same length down the column. VARCHAR may be a bit slower as it calculates each field down the column, but it saves on memory space. Which one to ultimately use is up to you.

Using both a CHAR and VARCHAR option in the same table, MySQL will automatically change the CHAR into VARCHAR for compatability reasons.

BLOB stands for Binary Large Object. Both TEXT and BLOB are variable length types that store large amounts of data. They are similar to a larger version of VARCHAR. These types can store a large piece of data information, but they are also processed much slower.

NUMBER TYPES

TINYINT()	-128 to 127 normal 0 to 255 UNSIGNED.
SMALLINT()	-32768 to 32767 normal 0 to 65535 UNSIGNED.
MEDIUMINT()	-8388608 to 8388607 normal 0 to 16777215 UNSIGNED.
INT()	-2147483648 to 2147483647 normal 0 to 4294967295 UNSIGNED.
BIGINT()	-9223372036854775808 to 9223372036854775807 normal 0 to 18446744073709551615 UNSIGNED.
FLOAT	A small number with a floating decimal point.
DOUBLE(,)	A large number with a floating decimal point.

DECIMAL(,) A DOUBLE stored as a string , allowing for a fixed decimal point.

The integer types have an extra option called UNSIGNED. Normally, the integer goes from an negative to positive value. Using an UNSIGNED command will move that range up so it starts at zero instead of a negative number.

DATE TYPES

DATE	YYYY-MM-DD.
DATETIME	YYYY-MM-DD HH:MM:SS.
TIMESTAMP	YYYYMMDDHHMMSS.
TIME	HH:MM:SS.

MISC TYPES

ENUM ()	Short for ENUMERATION which means that each column may have one of a specified possible values.
SET	Similar to ENUM except each column may have more than one of the specified possible values.

ENUM is short for ENUMERATED list. This column can only store one of the values that are declared in the specified list contained in the () brackets.

ENUM('y','n')

You can list up to 65535 values in an ENUM list. If a value is inserted that is not in the list, a blank value will be inserted.

SET is similar to ENUM except SET may contain up to 64 list items and can store more than one choice

ENVIRONMENT / OPERATING SYSTEM

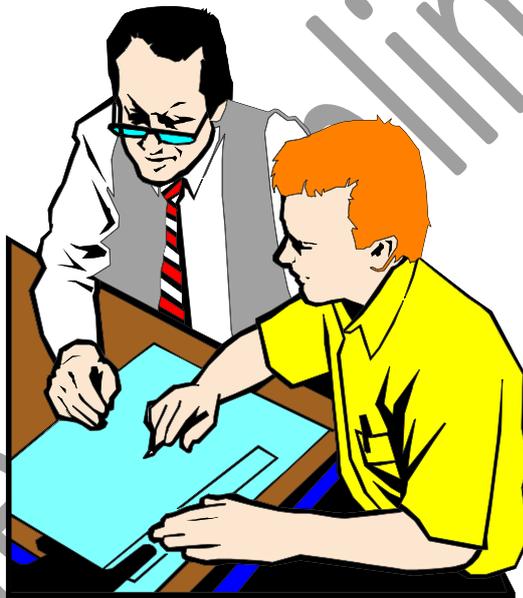
The age of standalone computing is long gone and it is now a world of networks. In this context, it is imperative for users to understand issues involved in working in a multi-user environment. Windows NT today is the most sought after Multi-user Operating System in the world. This is a 32-bit, powerful, user friendly and robust network operating system from Microsoft.

Windows NT is a big step up from Windows 95 both in terms of price and in terms of capabilities. NT Server is a well-built, fully functional, multi-user, multitasking operating system.

The release of NT Server 4 adds the popular Windows 95 user interface, the Internet Information Server (IIS) – capable of running Web, FTP, and Gopher services - a search engine called Index Server, the FrontPage HTML authoring package, and the Internet Explorer, all of which make Windows NT server a tough act to beat as a robust, capable web server.

The suggested Operating System for the server computer is Windows NT Server and for client computer is Windows 95.

SYSTEM STUDY



2.1 PRELIMINARY INVESTIGATION

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

- Preliminary investigation
- Determination of system requirements
- Design of system
- Development of software
- System testing
- Implementation, evaluation, and maintenance

A request to take assistance from information system can be made for many reasons, but in each case someone in the organization initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:

- 1) Request clarification
- 2) Feasibility study
- 3) Request approval

Request clarification: Many requests from clients and users in the organizations are not clearly defined, therefore it becomes necessary that project request must be examined and clarified properly before considering systems investigation.

2.2 SYSTEM DEVELOPMENT LIFE CYCLE

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject System Analysis and Design (SAD), mainly deals with the software development activities.

DEFINING A SYSTEM

A collection of components that work together to realize some objective forms a system. Basically there are three major components in every system, namely input, processing and output.

In a system the different components are connected with each other and they are interdependent. For example, human body represents a complete natural system. We are also bound by many national systems such as political system, economic system, educational system and so forth. The objective of the system demands that some output is produced as a result of processing the suitable inputs.

SYSTEM LIFE CYCLE

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

- System study
- Feasibility study
- System analysis
- System design
- Coding
- Testing
- Implementation
- Maintenance

Projecthelpline.in

The Different Phases Of Software Development Life Cycle Are Shown Below.

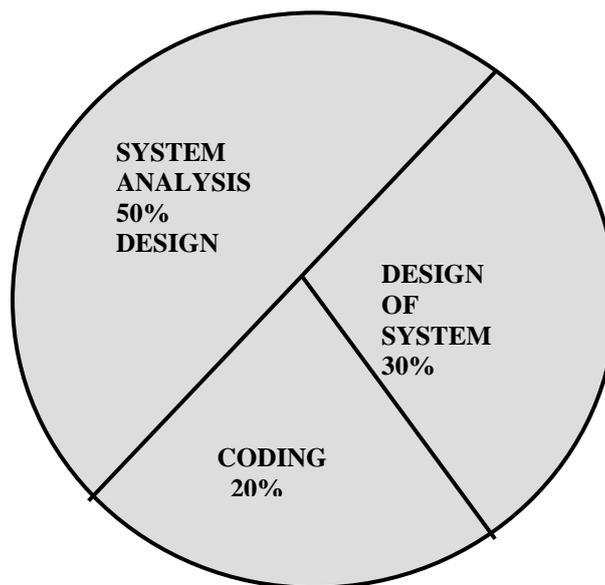
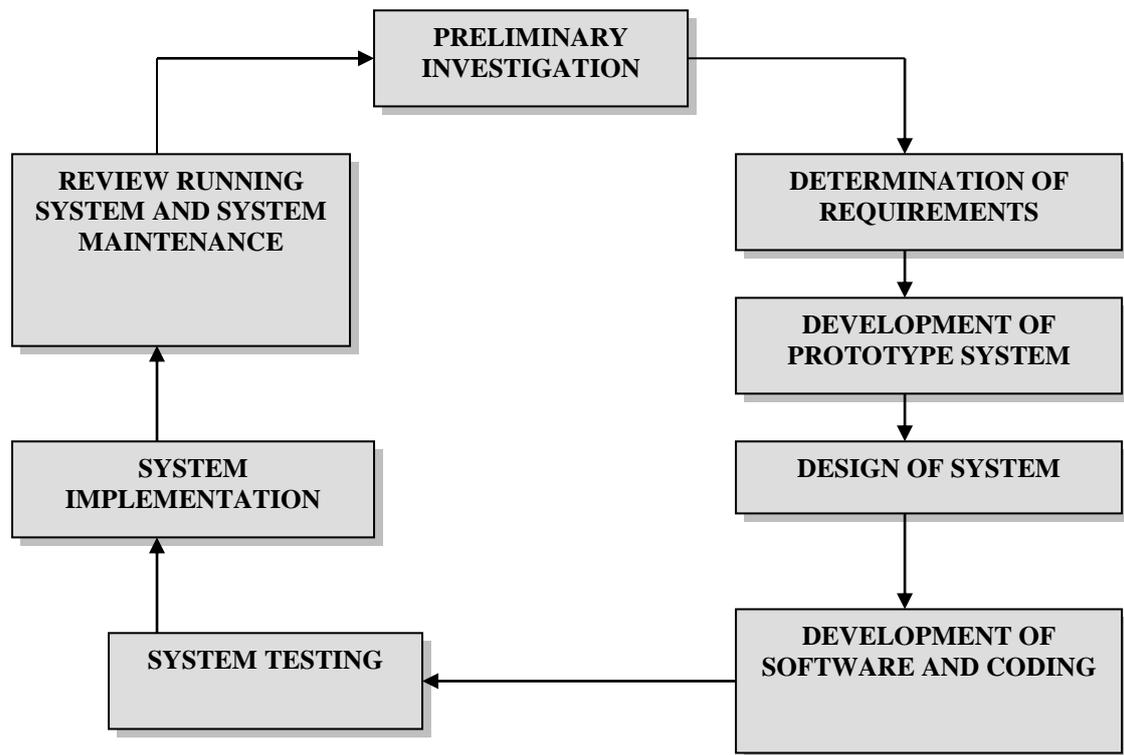


FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED

A system analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- ✓ What type of information is required?
- ✓ What are the constraints on the investigation?
- ✓ What are the potential problems that may make the task more difficult?

2.3 FEASIBILITY STUDY

The basic premise of system analysis is being done here. The primary goal of the system analysis stage is to identify problems and determine how they can be solved with the computer system. In formal SDLC methodologies, the first step in system analysis is feasibility study. A feasibility study is the quick examination of the problems, goals, expected cost of the system. The objective is to determine whether the problem can reasonably be solved with a computer system. In some cases, there may be a better alternative, or perhaps it is simply a short-term annoyance and will gradually disappear. In other cases, the problem may turn out to be more complex than was thought and involves users across the company. Also, some problems may not be solvable with today's technology. It might be better to wait for better technology. In any case, you need to determine the scope of the project to gain a better idea of cost, benefits, and objectives.

The feasibility study is typically written so that non-programmers can easily understand it. It is used to "sell" the project to upper management and as a starting point for the next step. Additionally, it is used as a reference to keep the project on track, and to evaluate the progress of the project team. Is the project cost-effective or is there a cheaper solution? Will the proposed system improve the operation of the bank; will complicating factors prevent it from achieving its goals? Does the technology exist and does the firm have the staff to make the technology work?

When the proposal is determined to be feasible, the team leaders are appointed and a plan and schedule are created. The schedule contains a detailed listing of what parts of the project are completed at each time. Of course, it is extremely difficult to estimate the true cost and completion dates. Nonetheless, the schedule is an important tool to evaluate the status of the project and the progress of the team.

Steps in feasibility Analysis are:

1. Identify deficiency by pinpointing, missing functions, unsatisfactory performance, and Excessive cost of operations.
2. Set goals to remove these deficiencies.
3. Goals must be quantified, realizable within the constraints of an organization, broken down into sub goals agreeable to all concerned.
4. Set goals not only to remove deficiencies but also to effectively meet competition. For instance, goals must be based on what competitors do.

2.4 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. More commonly known as cost / benefit analysis; in this procedure we determine the benefits and savings that are expected from a proposed system and compare them with costs. We found the benefits outweigh the costs; we take a decision to design and implement the new proposed system.

During the feasibility phase, broad alternatives solutions are examined. For each alternate solution the cost and benefits have to be examined before designing one of the alternatives.

Broad solutions will consist of:

1. Specifications of information to be made available by the system.
2. Description of what will be done manually and what the computer will do.
3. Specification of new computing equipment needed or specification of expansion of an existing computer.

➤ **COST AND BENEFIT ANALYSIS**

Developing an IT application is an investment. Since after developing that application it provided the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks because in some cases an estimate can be wrong and the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected cost.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

- i. Development cost** - Development costs is the cost that are incurred during the development of the system. It is one time investment.
- ii. Operating cost** - Operating cost are the expenses required for the day to-day running of the system. As, operating cost are wages, supplies and overheads.
- iii. Hardware/Software cost** - It includes the cost of purchasing or leasing of computes and it's peripherals. Software costs involves required software cost.
- iv. Personnel cost** - It is the money spent on the people involved in the development of the system.
- v. Facility cost** - Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lighting, and air-conditioning.
- vi. Supply cost** - These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and others.

➤ **BENEFITS**

We can define benefits as

Profit or Benefit = Income - Cost

Benefits can be accrued by:

Increasing income or

Decreasing costs or

Both

ITEM	COST (Rs.)
Computer	40,000 (depend upon the configuration)
Laser Printer	7,000
JSP	
MYSQL	
Scanner	4,000
Project cost	4,000 (approx.)
Total	55,000

Salary of on employee being around Rs.15,000. In one year annual salary of employee is equal to Rs. 80,000 and salary to four employees is Rs. 3, 20,000 for one year .From this analysis it can be seen that whatever money the companies will invest on making their system a computerized system will be recovered and half years of time (approx.). This analysis shows that how management is benefited by computerized system.

BENEFITS:

- 1) Fast and easy access to all Physical File Storage Management System related procedures and functions.
- 2) High level of security and authentication of each and every user.
- 3) Less need for personnel, thus, no monthly salaries, which leads to no extra funds.

4) Reliability is increased, as backups of files, and records can be made and saved in various different locations and information will be highly secure, unlike in file cabinets where entries can easily be ripped or tampered with by users.

5) The reception/front office will look much more neater and cleaner the environment they need, as there Won't be any cupboards or drawers which make the companis overcrowded.

6) There will be no longer the need for all the paper work required to make timely reports lists or other lists as the program generates then at anytime at a very quick pace.

2.5 TECHNICAL FEASIBILITY

Today, very little is technically impossible. Consequently, technical feasibility looks at what is practical and reasonable. Technical feasibility addresses three major issues:

1. Is the proposed technology or solution practical?
2. Do we currently possess the necessary technology?
3. Do we possess the necessary technical expertise, and is the schedule reasonable?

Is the Proposed Technology or Solution Practical?

The technology for any defined solution is normally available. The question whether that technology is mature enough to be easily applied to our problems. Some firms like to use state-of-the-art technology, but most firms prefer to use mature and proven technology. A mature technology has a larger customer base for obtaining advice concerning problems and improvements.

Do We Currently Possess the Necessary Technology?

Assuming the solution's required technology is practical, we must next ask ourselves, is the technology available in our information systems shop? If the technology is available, we must ask if we have the capacity. For instance, will our current printer be able to handle the new reports and forms required of a new system?

If the answer to any of these questions is no, then we must ask ourselves, Can we get this technology? The technology may be practical and available, and, yes, we need it. But we simply may not be able to afford it at this time. Although this argument borders on economic feasibility, it is truly technical feasibility. If we can't afford the technology, then the alternative that requires the technology is not practical and is technically infeasible!

Do We Possess the Necessary Technical Expertise, and Is the Schedule Reasonable?

This consideration of technical feasibility is often forgotten during feasibility analysis. We may have the technology, but that doesn't mean we have the skills required to properly apply that technology. For instance, we may have a database management systems (DBMS). However, the analysis and programmers available for the project may not know that DBMS well enough to properly apply it. True, all information systems professionals can learn new technologies. However, that learning curve will impact the technical feasibility of the project; specifically, it will impact the schedule.

As mentioned earlier, the current operational state of Physical File Storage Management System is very primitive as all storage is done on hand written database. These files are then placed in drawers or cabinets and tagged in a sorted order. The Physical File Storage Management System contains over a large number of drawers and cabinets. Each cabinet takes a large space. Other than this it uses many kind of papers to calculate and maintain different account works. Since there has been no use of computer in the Physical File Storage Management System so far, for storage of data, there is no currently used software.

The software & hardware to run my project, and is it's usage is given in the table below:

2.6 OPERATIONAL FEASIBILITY

It is mainly related to human organizational and political aspects. The points to be considered are:

- o What changes will be brought with the system?
- o What organizational structures are disturbed?
- o What new skills will be required? Do the existing staff members have these skills?
- o If not, can they be trained in due course of time?

Generally project will not be rejected simply because of operational infeasibility but such considerations are likely to critically affect the nature and scope of the eventual recommendations.

For operational feasibility study we appointed a small group of people who are familiar with information system techniques, who understand the parts of the business that are relevant to the project and are skilled in system analysis and design process.

2.7 FEASIBILITY REPORT

After studying the feasibility of the project we came to the following points, these results may change according to further analysis and design.

PROJECT NAME: PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

DEFINITION OF PROBLEM OR OPPORTUNITY: We have to make a computerized system (software) to make the working of Company easy and efficient to handle the new projects and old projects status so that software will replace the manual work with automated computerized process.

EXPECTED BENEFITS:

- Reduce the number of employee.
- Save money.
- Increase the efficiency of workers.

- Reduce the response time.
- Improve the service quality.
- Reduce the bulk of paper work.
- Reduce the chance of error by human.
- Increase the accuracy in result.

Projecthelpline.in

SYSTEM ANALYSIS



3. 1 IMPORTANCE OF COMPUTERIZED

There are several attributes in which the computer based information works. Broadly the working of computer system is divided into two main groups:

- ◆ Transaction System
- ◆ Decision Support System

Transaction System:

A transaction is a record of some well-defined single and usually small occurrence in a system. Transactions are input into the computer to update the database files. It checks the entering data for its accuracy. This means that numeric data appears in numeric field and character data in character field. Once all the checks are made, transaction is used to update the database. Transaction can be inputted in on-line mode or batch mode. In on-line mode, transactions are entered and updated into the database almost instantaneously. In batch mode, transactions are collected into batches, which may be held for a while and inputted later.

Decision Support System:

It assists the user to make analytical decision. It shows the various data in organized way called analysis. This analysis can be made to syrdy preferences and help in making decisions.

Computer system works out best with record maintenance. It will tell you which customer would get how much pending/reports statements. It will also help to search the information about a particular person by simply entering his telephone number. User can store information as per requirement, which can be used for comparison with other reports.

3.2 PRINCIPLES OF SYSTEM ANALYSIS

Principles:

1. Understand the problem before you begin to create the analysis model.
2. Develop prototypes that enable a user to understand how human machine interaction will occur.
3. Record the origin of and the reason for every requirement.
4. Use multiple views of requirements like building data, function and behavioral models.
5. Work to eliminate ambiguity.

A Complete Structure:

The limited time and resources have restricted us to incorporate, in this project, only the main activities that are performed in news sites, but utmost care has been taken to make the system efficient and user friendly.

For the optimum use of practical time it is necessary that every session is planned. Planning of this project will include the following things:

- Topic Understanding.
- Modular Break – Up of the Syst
- Processor Logic for Each Module.
- Database Requirements.

Topic Understanding:

It is vital that the field of application as introduced in the project may be totally a new field. So as soon as the project was allocated to me, I carefully went through the project to identify the requirements of the project.

Modular Break –Up of the System:

- Identify The Various Modules In The System.
- List Them In The Right Hierarchy.
- Identify Their Priority Of Development

➤ Description Of The Modules:

3.3 System Analysis

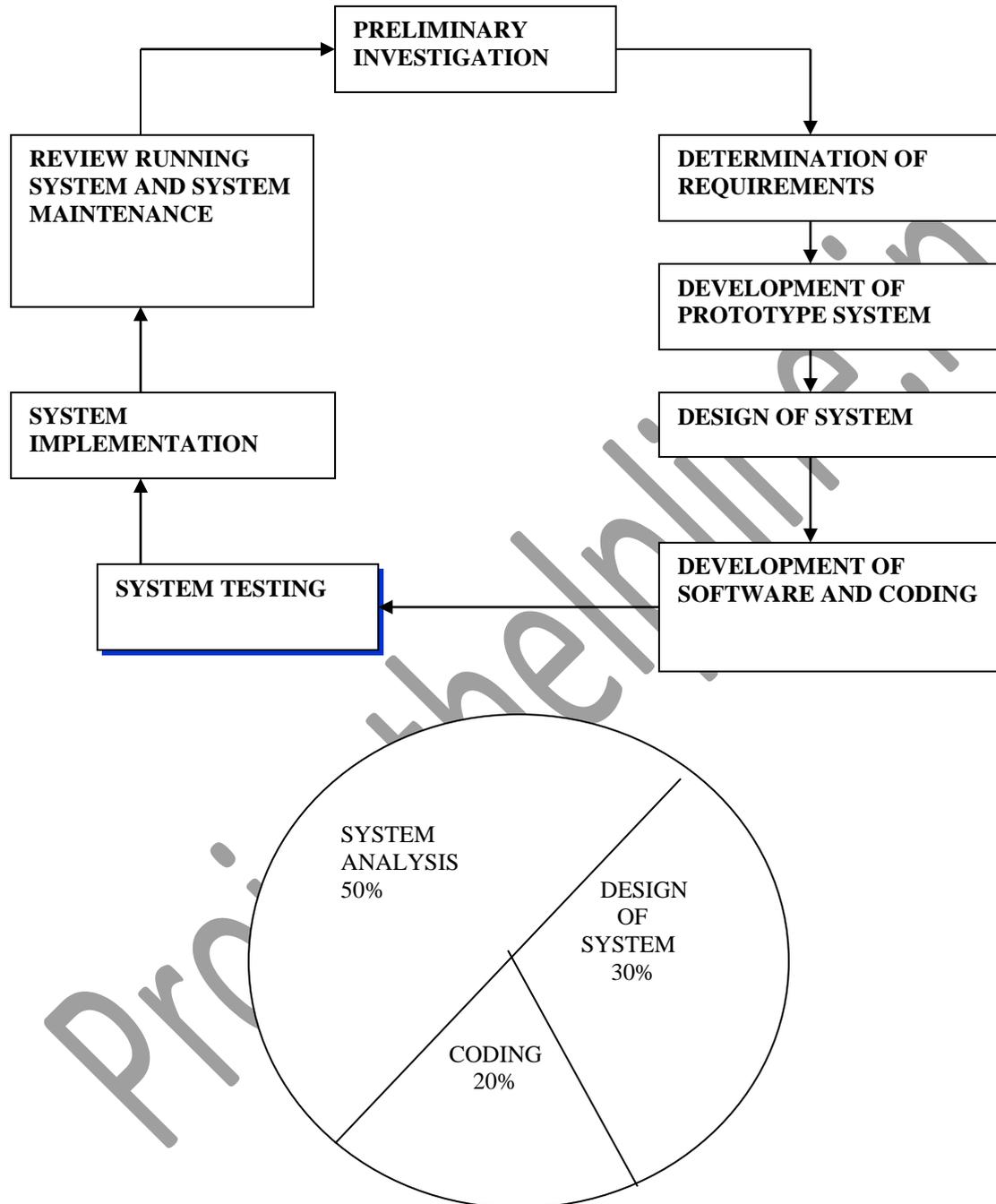


FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED

A system analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system.

One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

- What type of information is required?
- What are the constraints on the investigation?
- What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system, the total system was designed and can be described as under:

The three major parts of the system are:

➤ **Providing Information:**

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide access to records of various products with quick update to latest modifications in the records. This thing is not at all possible in printed material, which are updated only once a few weeks. It also gives

information about the general usage of the system for first time visitors. The system itself works as a information provider for Software Testing.

Alert when available: Through the survey it was clearly that there is a need to device an alternative way for providing alert facility to the user. Sometimes the product which customer demand is not available at that moment, user can register demand of customer and when its available, system gives an alert to the user that customer had registered a customer request with the same match.

Constraints: After the objectives were clear during the analysis phase, it was essential to understand the constraints in order to plan and avoid problems arising during detailed analysis.

Technology - the customer may be committed to a particular hardware or software solution. The software required in this case is: complete Java developer kit, Microsoft windows environment for MS - access.

Budget - if budget is a real constraint, the budget of the new system proposed would be constantly compared with that of the existing system or any Alternatives solution. In this case during the economic feasibility study it has been clearly proved that the new system is definitely more feasible than the alternative solution possible. Organization must implement a system which saves the effort, also its provide an easy method for customer who investigate each detail itself.

Scope - what is the area under investigation in this project? What are the boundaries of the system? What is the extent of possible usage of the new system?

More and more people are now having access to organization and watch independently Details of new upcoming stock. Hence the scope is constantly increasing. However its usage can be increased many folds with a little investment from the organization side by implanting touch screen computer kiosks at various convenient positions at the service station.

Environmental Analysis- The external entities for an organization are its Supplier's customers or any individual.

3.4 METHODS USED FOR GATHERING INFORMATION

This project follows the waterfall model in software engineering for implementation. The Waterfall Model was first Process Model to be introduced. It is also referred to as a linear-sequential life cycle model and often used in software development processes. It is very simple to understand and use. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases.

In "The Waterfall" approach, the whole process of software development is divided into separate phases. In Waterfall model, typically, the outcome of one phase acts as the input for the next phase sequentially. All these phases are cascaded to each other in which progress is seen as flowing steadily downwards (like a waterfall) through the phases of requirement, analysis, design, construction, testing, deployment and maintenance. The next phase is started only after the defined set of goals are achieved for previous phase.

The methods used for gathering information about the existing information system are as followed.

- (a) Review of records.
- (b) Observation of the functioning system.
- (c) Interviews.
- (d) Questionnaires.

In order to create an informative and practical system, a system analyst would have to have some kind of way to view the current system. Receiving feed back on what can be done to improve the current system, and how much the current system is acceptable to the users.

1) Requirement analysis: -

The main part of problem is to obtain a clear understanding of the needs of user and what exactly are desired from the software. It is used for specifying the requirement.

Fact finding tools:

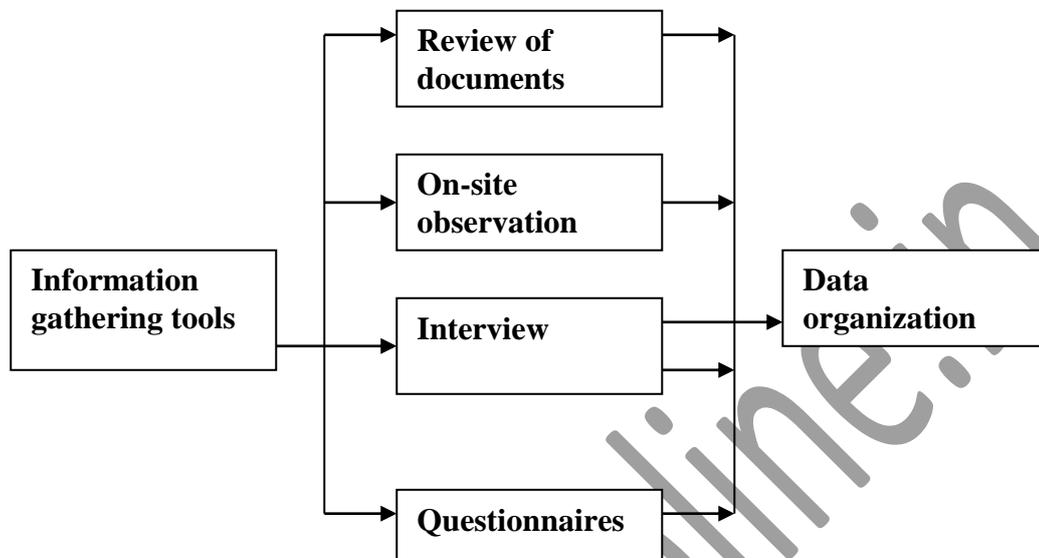
After obtaining the background knowledge, I began to collect data on the existing system's output, input, and costs. The tools used in data collection / information gathering are:

Review of the written Documents

On-site Observation

Interviews

Questionnaires



Review of the written documents:

In this phase we analyzed all the documents like the day books, supply report, order generating forms, supply forms, account etc. All these things describe the format and functions of the current system included in most manuals are system requirement that help determine how III various objectives are met.

The form is one of the most important source through which ! draw some conclusion like:

1. Who use the form(s)? How important are they to the user?
2. Do the forms include all the necessary information? What item should be added or deleted?
3. How readable and easy to follow is the form?
4. How does the information in the form help other user make better decision?
5. What other uses does the form offer the user area?

By analyzing all the details we draw a conclusion that what are the merit and De-merit of the current phase. Will the company contain all the back up of all the important document of not sales person contains all the information about the available vehicles or not. But above all there are some problems with the on site observations that one analyst must face during analysis like:

1. Take long time and get inefficient result
2. Attitude and motivation of subject cannot be readily observed
3. Observation are subject to error
4. In a complex situation it can be very time consuming

So for this we switched towards the other fact finding tools like interviews and questionnaires.

3.5 INTERVIEWS AND QUESTIONNAIRES:

The interviews is a face to face interpersonal role situation in which a person called The interview asks a person being interview questions designed to gather information about a problem area. The interview is the oldest and most often used device for gathering information in systems work. It has qualities that behavioral and on-site observation do not possess .it can be used for two main purposes:

- (a) as an exploratory device to identify relations or verify Information
- (b) to capture information as it exists

1. On site observation:

It is not the easy task to do. In the on site observation the main objective is to get close as possible to the real system that are being studied. There are some questions that can serve as a guide for the on site observations:

1. What kind of the system is it? What does it do?
2. Who runs the system? Who are the important people in it?
3. What is the history of the system? How it get to its present stage of the Development first in this phase we more likely listened than talk and to and to listen with a sympathetic and genuine interest when information is conveyed. We have not to give any advice or passing

moral judgment on what is observed. Not to argue with anyone or not to show any hostility towards one person and undue friendliness towards another.

So in the on site observation we first visited to the entire organization watched all the transactions, watched the way of taking orders by the sales executive. What types of forms are used in the entire place? Are all forms are written, printed or not.

There are four primary advantages of the interviews:

1. its flexibility
2. easy to validate the information gathered
3. effective technique for elicit information about complex subject
- 4 instance, goals must be based on what competitors do.

Questionnaires:

It is usually associated with the self-administrated tools with items of the closed or fixed alternative type. By this nature a questionnaires offer the following advantages:

- (a) It is economical and requires less skill to administer than the interview
- (b) Unlike the interview which generally questions one subject at time questionnaire can be administrated to large number of individuals simultaneously.
- (c) The questionnaires place less pressure on the subjects for immediate responses. Respondents have to think the question over and do calculations to provide more accurate data.

Mainly interviews and questionnaires are divided into two categories.

1. The unstructured alternative
2. The structured alternative

The unstructured interview is a relatively nondirective information technique. The role of the analyst as an interviewer is to encourage the respondent to talk freely and serve as a catalyst to the expression of feeling and opinions.

In the structured alternative approach the questions are presented with exactly the same wording and in the same order to all subjects. Questions may be closed or open ended. An open-ended question requires no response direction or the specific response.

An analyst asks the questions at each level of management either it is top level, middle level or bottom level and at each department in the client's site and at the manufacturer.

Following are the sample questions which we asked in the form of questionnaires and interviews:

Samples Questions asked to Computer Operators

(of similar kind of Centers with automated billing systems)

- 1.) Are you satisfied with the current system?
i. YES ii. NO iii. Partially Satisfied
- 2.) On an average how much customers do you get daily?
i. >100 ii. <100 iii. Can't say
- 3.) Are you satisfied with the facilities available to you?
i. Yes ii. NO iii. Partially Satisfied
- 4.) What type of crowd is there mostly?
i. Students ii. Working Personals iii. General (both)
- 5.) Who are your regular customers?
i. Students ii. Working Personals iii. Not specific
- 6.) Which kind of working environment do you have?
i. Crowdy ii. Vibrating iii. Can't Say
- 7.) Which Platform do you work on?
i. GUI Based ii. Non-GUI Based iii. Don't Know

8.) Which Database is implemented for the File System?

- i. MS ACCESS ii. MY SQL iii. ORACLE
- iv. Others (Please Specify)

9.) Any Suggestions for further improvement?

Samples Questions asked to Administrators

(Of similar kind of organizations with automised billing systems)

1.) Are you satisfied with the current system?

- i. YES ii. NO iii. Partially Satisfied

2.) Are you satisfied with the work done by your staff?

- i. YES ii. NO iii. Partially Satisfied

3.) What type of up-gradation do you intend to have in the current system?

- i. Technical ii. Manual iii. Monetary

4.) Which type of promotional schemes you are coming up?

- i. Revise discount rates ii. Add more facilities for customers iii. None

5.) What do you emphasize on (Regarding type of work done at the center) ?

- i. Quality ii. Quantity iii. Both.

6.) What is the Price/Profit ratio in Off and On season?

- i. Positive ii. Negative iii. Constant/ Average

7.) Are you satisfied with the number of customers coming here?

- i. Yes ii. NO iii. Can't say.

8.) Which Platform do you work on?

- i. GUI Based ii. Non-GUI Based iii. Don't Know

9.) Which Database is implemented for the File System?

- i. MS ACCESS
- ii. My SQL
- iii. ORACLE
- iv. Others (Please Specify)

10.) How much security is needed for the implementation of atomization?

- i. Very High
- ii. Average
- iii. Can't Specify

11.) Any Suggestions for further improvement?

Sample Questions asked to consumers (Center's customers)

Personal Information:

- i. NAME.
- ii. Customer Type ->
 - a.) School Student
 - b.) Collage Student
 - c.) Working Personal

1.) How do you like the facilities provided by the Center?

- i. Good
- ii. Betters than others
- iii. Best

2.) Are you satisfied with the Quality of service?

- i. Yes
- ii. No
- iii. Partially Satisfied.

3.) What do you think about the discount offered?

- i. Too Less
- ii. Appropriate
- iii. Should be increased

4.) Is the Price of auto items adequate?

- i. Yes
- ii. No
- iii. Can't Say

6.) Does the organization need further improvements?

- i. Yes
- ii. No
- iii. Can't say

Answers we got from various computer operators

- (i) 45% answered
- (ii) 35% answered
- (iii) 20% answered

Answers we got from various administrators

- (i) 42% answered
- (ii) 34% answered
- (iii) 26% answered

- i - positive answers
- ii - Average answers
- iii - Negative Answers

Answers we got from various consumers

- 34% answered
- 36% answered
- 30% answered

Projecthelpline.in

SYSTEM DESIGN

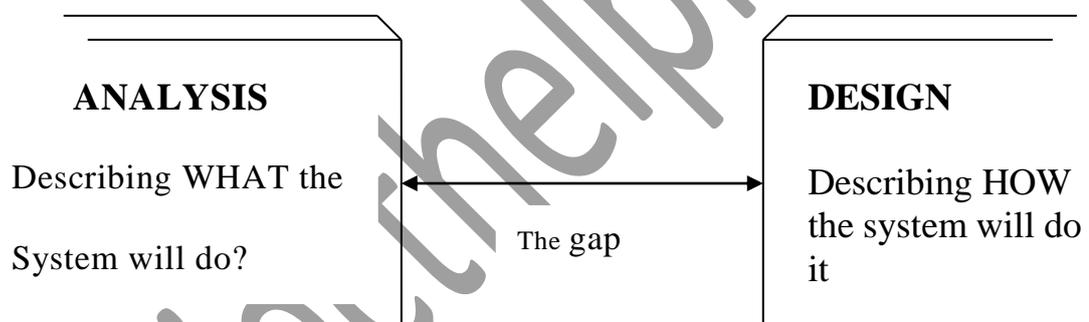
Projecthelping.com

The final deliverable from system analysis is a document containing an unambiguous statement of the client's requirements from the new system and what the development project will have to deliver in order to be considered a success.

The functional specification is the starting point for designing, which depends to a large extent on the accuracy and thoroughness with which the analysis has been carried out. Understanding of the business, appreciation of the client's problems and documentation of requirements provide the foundation on which the designing is based.

A key factor in this approach is the use of the structured techniques.

Analysis ends with a description of what the new system must do to fulfill the requirements of the organization, while design specifies how this will be done by selecting one of the many ways of doing it.



The structured techniques used during the analysis which provide this logical view are:

- Data flow diagrams - representing the process, which manipulate the data as it passes through the system.
- Entity model - showing the relationship within the data items held within the system.
- A data dictionary - providing an overall consistent definition of the data used during the system development. This definition includes the content of the data stores, data flows and the process shown on the data flow diagrams, and the entities that make up the entity model.

4.1 DESIGN OBJECTIVES AND CONSTRAINTS

Flexible –

The design would enable future requirements of the organization to be incorporated without much difficulty. Often the organizational needs and objectives change over time and hence such a design enables the system to reflect these changes.

• Maintainable -

A good design is easy to maintain and this reduces the client's maintenance cost, which usually represents a proportion of the lifetime of the system.

• Portable -

A client for whom the software was developed may wish to change the hardware on which the system run. A good design is portable - in other words it is capable of being transferred from one machine environment to another with minimum amount of effort.

• Easy to use -

With increasing number of general users having exposure to computers and access to web sites, expectations of computer applications in term of their ease of use are also increasing. A good design will result in a system which is 'user - friendly' - easy to understand, not difficult to learn how to use and straightforward to operate.

• Reliable -

The system designed must be secure against human error, deliberate misuse or machine failure, and which the data will be stored without corruption.

4.2 PHYSICAL DESIGN

The design phase focuses on the detailed implementation of the system recommended in the feasibility. Emphasis is on translating performance specifications into design specifications. The design phase is a transition from user-oriented document to a programmer-oriented document.

4.2.1 Design Methodology:

Design Methodology is a way to transform the "art" of system analysis and design into an "engineering - type" discipline. It explains the relationship amongst various modules and programs within the system. It standardizes the approach to analysis and design, simplifies design by segmentation, improves documentation and subsequent maintenance and enhancements.

The following structured diagram can appropriately represent the relationship between various modules.

4.2.2 Design Overview:

In analyzing the present system a great deal of information was collected during the investigation and feasibility phases through list of problems and requirements, interview reports, questionnaires, onsite observations, manuals and determining potential solutions.

It is important to record this information in an unambiguous, concise manner which will be clear and accessible to others, and which can be used by other analysts and designers involved in developing the system. Structured techniques help us to record the information in this way, using diagrams and minimum amount of the text.

Structured analysis is a set of techniques and graphical tools that allow the analyst to develop a new kind of system specification that are easily understandable to the user. The traditional approach of organizing data through flowcharts support future developments and simplify communication with the user but focus on the cost/benefit and feasibility analysis, project management, hardware and software selection, and personal considerations. In contrast, structured analysis considers new goals and structured tools for analysis, which provide the basis for design and implementation.

4.2.3 Process Modeling:

System design goes through two phases of development: logical and physical. Logical implementation represented by Data Flow Diagram shows the logical flow of a system and defines the boundaries of the system it describes the input (source), outputs (destinations), data bases (data stores), and procedures (data flows) - all in the format that meets the user's requirements. The logical implementation of the whole project can be represented as under through Data Flow Diagrams (DFD).

4.3 DATA FLOW DIAGRAM

Data flow diagrams are the most commonly used way of documenting the processing of the candidate system. As their name suggest they are a pictorial way of representing the flow of data into, around, and out of the system. They are easily understandable and are less prone to misinterpretation than textual description. A complete set of DFDs provides a compact top - down representation of the system, which makes it easier for the user and the analyst to envisage the system as a whole.

DFDs are constructed using four major components:

- **External entities** - represents the sources of the data that enter the system or the recipients of the system that leave the system. For example - passenger is the usual receiver of information and supplier of data during form filling.
- **Data stores** - represent the stores of the data within the system example: computer files, databases or in the manual system files, etc. data stores can not be linked directly by data flows either to each other or to external entities without an intervening process to transform them.
- **Processes** - represent activities in which data is manipulated by being stored or retrieved or transformed in some way. Process names are generally unambiguous and convey as much meaning as possible without being too long. Example: verify data, acquired time schedule etc.

Data flows - represents the movement of data between other components.

4.4 DATA MODELING

Store Room Employee Module:

This is the main module in the proposed project. The Store room staff or employee can read and write information about any customer files. The Store room staff can also update, create and update the record of customer files as requirement and implementation plan.

Customer Module:

This module contains the customer files records. User can retrieve the details of customer file and customer can also generate track report of customer confidential file.

Administrator Module:

This module creates employee of staff and customer account and this module contains the customer files records. Administrator can retrieve the details of customer file and employee file, customer feedback, file details can also generate track report of customer confidential file.

Data modeling defines primary data objects, composition of each data object, and attributes of the object, relationships between each object and other objects and between objects and the processes.

Data Objects: A data object is a representation of almost any composite information that must be understood by the software. By composite information, we mean something that has a number of different properties or attributes. A data object encapsulates data only there is no reference within a data object to operations that act on the data.

Attributes: Attributes define the properties of a data object and take on one of three different characteristics. They can be used to:

1. Name an instance of data object.
2. Describe the instance.
3. Make reference to another instance in other table.

Database Design

The PFSMS database consists of 8 tables. Each and every table is setup with the proper Integrity constraints to work properly. The structure of the database tables is given below:

Table: Admin

Field Name	Field Type	Constraints	Description
Username	Varchar (30)	Not Null	Stores the username of administrator
Password	Varchar (30)	Not Null	Stores the password of administrator

Table: Employee

Field Name	Field Type	Constraints	Description
Emp_id	Varchar(30)	Primary key	Stores Employee id
Username	Varchar(30)	Not Null	Stores Username
Password	Varchar(30)	Not Null	Stores Password
Name	Varchar(30)	Not Null	Stores Name
Designation	Varchar(30)	Not Null	Stores Designation
Department	Varchar(30)	Not Null	Stores Department
Email_ID	Varchar(30)	Not Null	Stores Email id
Phone_no	Varchar(30)	Not Null	Stores Phone

Table: Customer

Field Name	Field Type	Constraints	Description
ID	Varchar(11)	Not Null	Store Customer Id
Username	Varchar (100)	Not Null	Stores Username
Password	Varchar (100)	Not Null	Store password
Name	Varchar (100)	Not Null	Stores Name
Cname	Varchar (100)	Not Null	Stores Company Name
Address	Varchar(100)	Not Null	Store Address
City	Varchar(100)	Not Null	Store City
State	Varchar(100)	Not Null	Store State
Email	Varchar(100)	Not Null	Store Email
Phone	Varchar(100)	Not Null	Store Phone

Table: Feedback

Field Name	Field Type	Constraints	Description
Id_no	Varchar (30)	Not Null	Store feedback id
Name	Varchar (30)	Not Null	Store name
Email_id	Varchar (30)	Not Null	Store Email id
Phone_no	Varchar (30)	Not Null	StorePhoneNumber
State	Varchar (30)	Not Null	Store State
comment	Varchar (30)	Not Null	Store comment

Table: file_detail

Field Name	Field Type	Constraints	Description
Id	Varchar (100)	Not Null	Store File id
File_Name	Varchar (100)	Not Null	Store File name
File_type	Varchar (100)	Not Null	Store file type
Flocation	Varchar (100)	Not Null	Stores File location
customer_id	Varchar (100)	Not Null	Stores Customer id
Desc	Varchar (100)	Not Null	Stores Description
EntryDate	Varchar (100)	Not Null	Stores File Date
Status	Varchar (100)	Not Null	Stores status of file

Table: Issue

Field Name	Field Type	Constraints	Description
id	int (11)	Not Null	Store Issue id
Fid	Varchar (100)	Not Null	Store File id
Fname	Varchar (100)	Not Null	Store File name
Ftype	Varchar (100)	Not Null	Store File Type
Empid	Varchar (100)	Not Null	Store Employee Id
Status	Varchar (100)	Not Null	Stores status of File

Relationships: Data objects are connected to one another in a variety of different ways. We can define a set of object relationship pairs that define the relevant relationships.

□ **CARDINALITY AND MODALITY:**

❖ **Cardinality:**

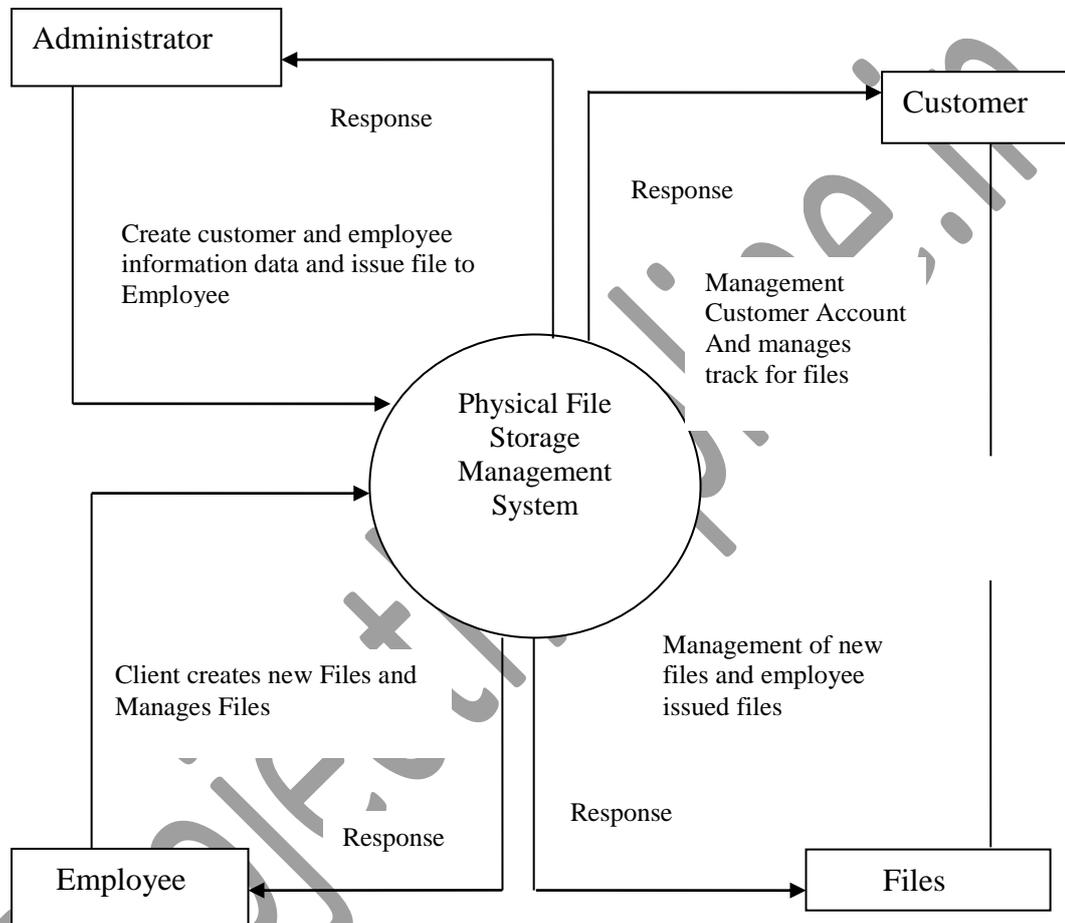
The data model must be capable of representing the number of occurrences of objects in a given relationship. The cardinality of an object relationship pair is

- ◆ **One-To-One (1:1):** An occurrence of object 'A' can relate to one and only one occurrence of object 'B' and vice versa.
- ◆ **One-To-Many (1:N):** One occurrence of object 'A' can relate to one or many occurrences of object 'B' but an occurrence of object 'B' can relate to only one occurrence of object 'A'.
- ◆ **Many-To-Many (M:N):** An occurrences of 'B' and an occurrence of 'B' can relate to one or many occurrence of 'A'.

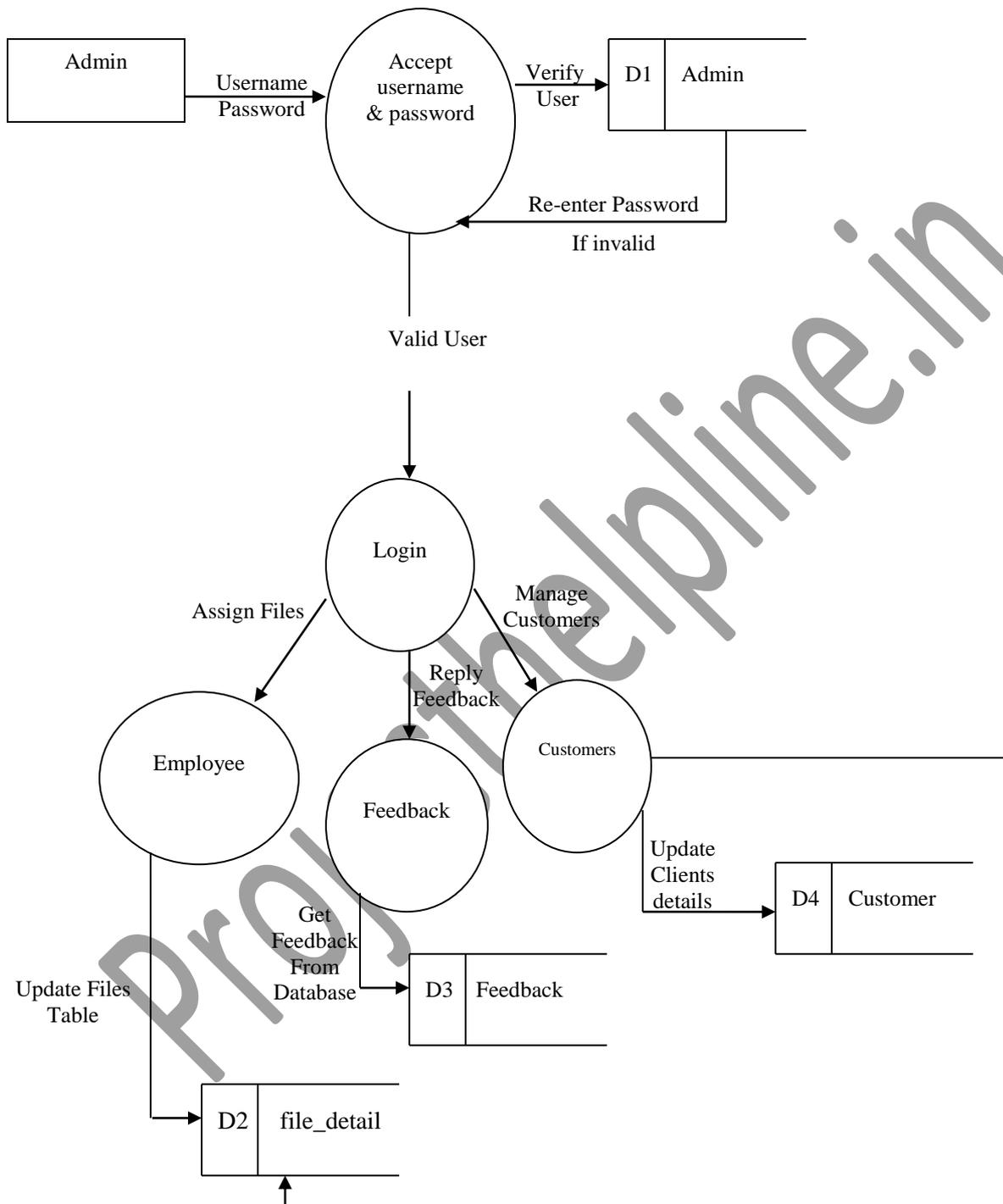
❖ **Modality:**

The modality of a relationship is zero if there is no explicit need for the relationship to occur or the relationship is optional. The Modality is one if the occurrence of the relationship is mandatory.

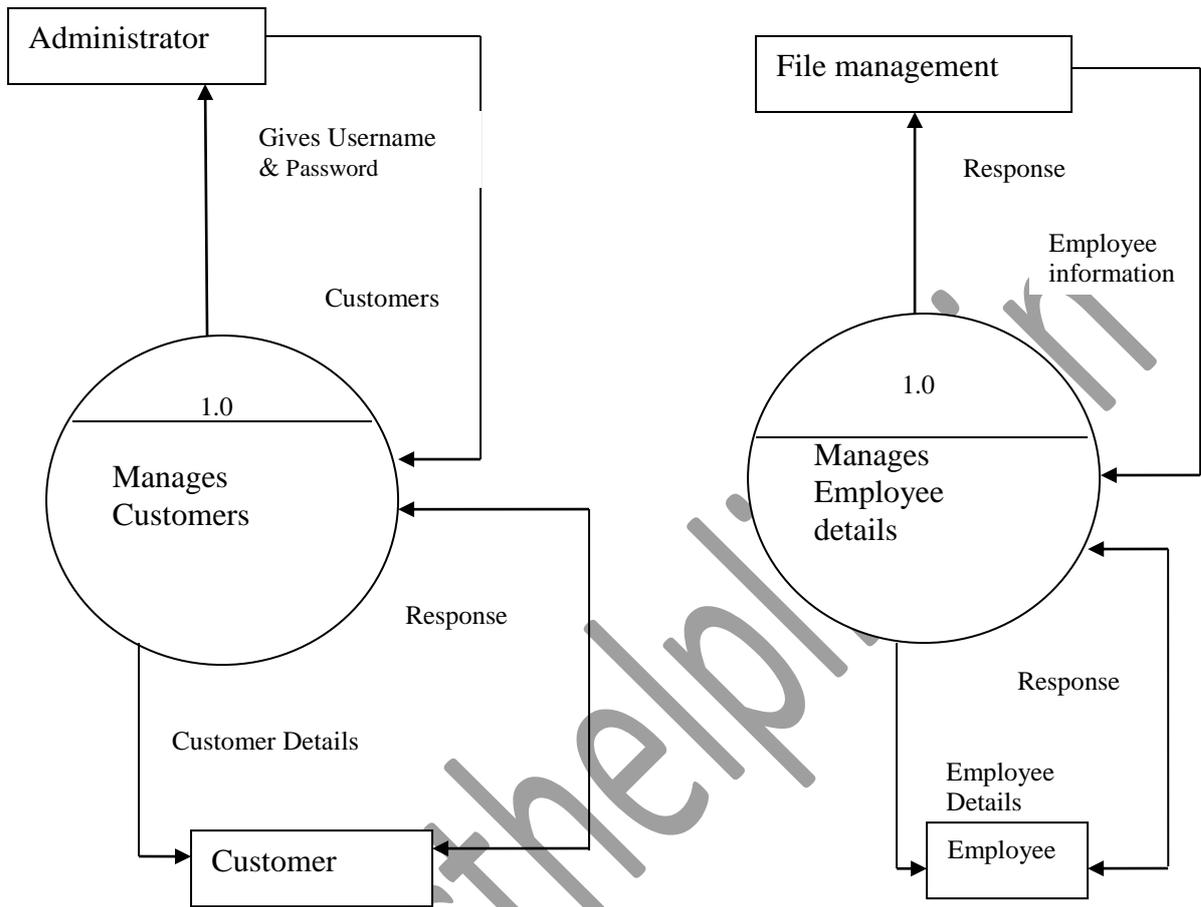
4.6 CONTEXT LEVEL DFD FOR
PHYSICAL FILE STORAGE MANAGEMENT SYSTEM



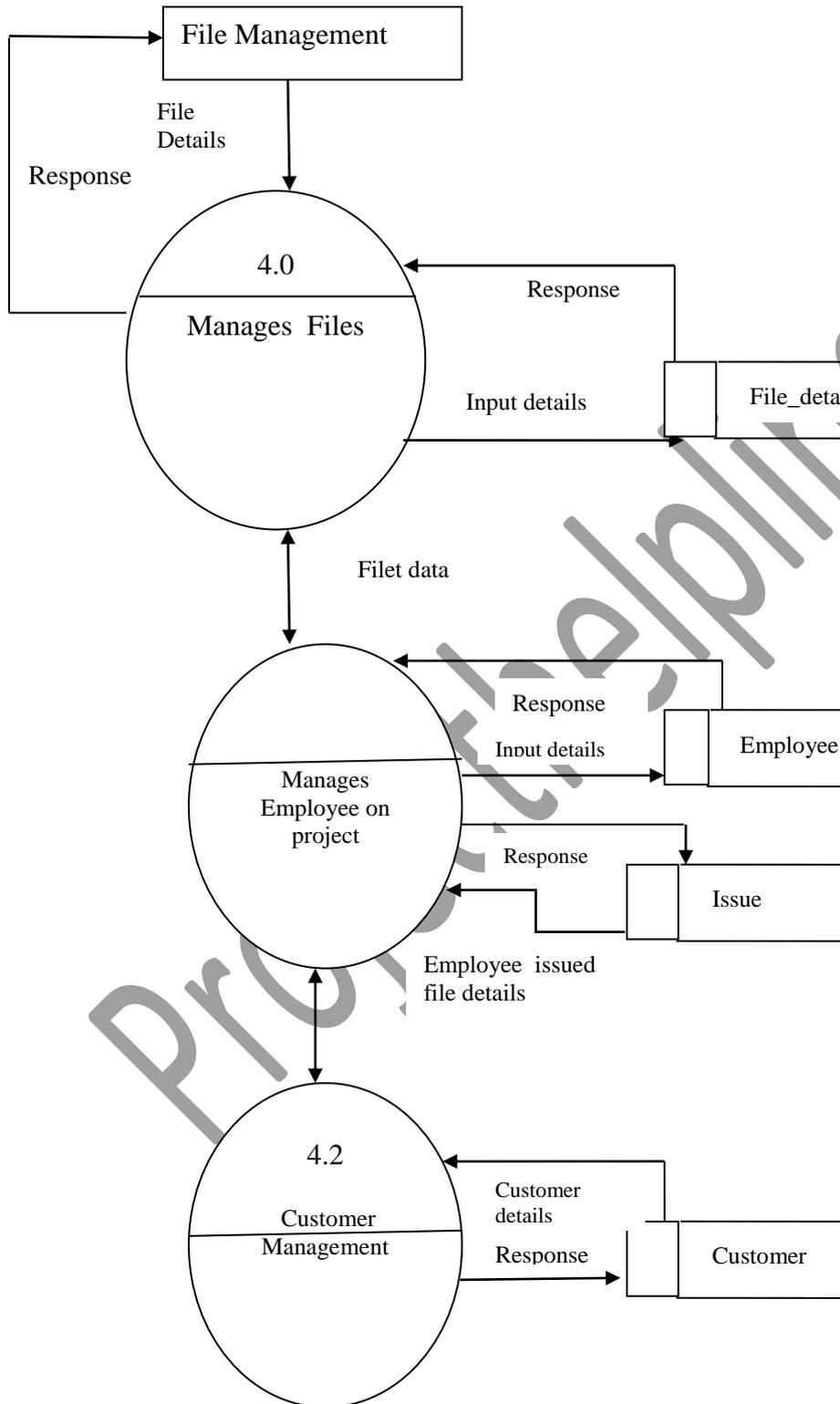
1st Level DFD For Physical File Storage Management System



IIInd Level DFD Administrator Module



IIRD Level DFD Physical File Storage Management System



4.7 MODULES OF THE PROJECT

Complete Structure of the Project

Modules and their description PFSMS comprises of four modules. Brief description of each and every module is given below:

- (i) **Administrator Module:** This module will be responsible for managing of the new Files and Issue new files and update status assigned to employee. In PFSMS administrator can create customers and employees. Administrator manage customer report, employee report and file reports with status along with file name, file type, file location. Customer Details and Employee details can be managed by this module.
- (ii) **Customer Management Module:** This module will be responsible for checking his file status and can track for file that are assigned or completed. The customers who will use the PFSMS according to store his files.
- (iii) **Employee Management Module:** This module will be responsible for creating files, maintaining the files status along with file name, file type, file date and file status so that easy deployment of files can be possible.
- (iv) **Files Management Module:** This module will be responsible for handling the different customers file, taking new files from customers. The module is solely responsible for managing new file, issued file details and balancing the load factor between the customers, employees and administrator.

Different modules that makeup these systems are briefly described below:

Module 1: Password Module

In this module, User enters a password and the software checks its validity. If the password is valid then option is given to change the password, otherwise “Invalid User/Password” message is displayed.

Module 2: Creating new Entity (Files, Customers, Employees, Administrator)

In this module, whenever a new entity is required to be added the corresponding forms are opened and the database is manipulated to check whether the data is already existing or not.

If it already exists, then it prompts that “Entry already existing” and if not than the data is entered with the various validation checks.

Module 3: Modifying/Updating Existing Entities (File Status, Files Issued to Employee and File Status)

In this module, whenever an existing entity is required to be modified the corresponding forms are opened and the database is manipulated and the data is fetched. Now the user can made the required changes and then accordingly, he updates the data. Again, the checks are followed in case there is any invalid entry.

Module 4: File Report

This function keeps track of file records. When this record is checked by the concerned person then the file report is generated for the customers. In this function, file report is generated according to the information's given by the employess in File Created Form. File Status are updated by work. Head of the Department generate file track report according to these information's.

Module 5: Issue: This field will represent the file issue record within a particular application the customer want to create the defect for.

Module 6: Feedbacks

In this module, Feedback details are given by the users and checked by the administrator of the site.

4.8 PROCESS LOGIC FOR EACH MODULE:

Process Logic of Module is a tool that may be useful in planning and evaluating projects. Our logic model contains four components with Inputs-Outputs-Outcomes being central to the built in response to the model:

- **Situation:** The context and need that gives rise to a file or initiative; logic modules are built in response to an existing situation.
- **Inputs:** The resources, contributions, and investments that are made in response to the situation. Inputs lead to Outputs.
- **Outputs:** The activities, products, methods, and services that reach people and users. Outputs lead to outcomes.
- **Outcomes:** The results and benefits for individuals, groups, agencies, communities and/or systems.

4.9 SCHEDULING

Scheduling of a software project does not differ greatly from scheduling of any multi- task engineering effort. Therefore, generalized project scheduling tools and techniques can be applied with little modification to software projects.

Program evaluation and review technique (PERT) and critical path method (CPM) are two project scheduling methods that can be applied to software development. Both techniques are driven by information already developed in earlier project planning activities.

Estimates of Effort

- A decomposition of the product function.
- The selection of the appropriate process model and task set.
- Decomposition of tasks.

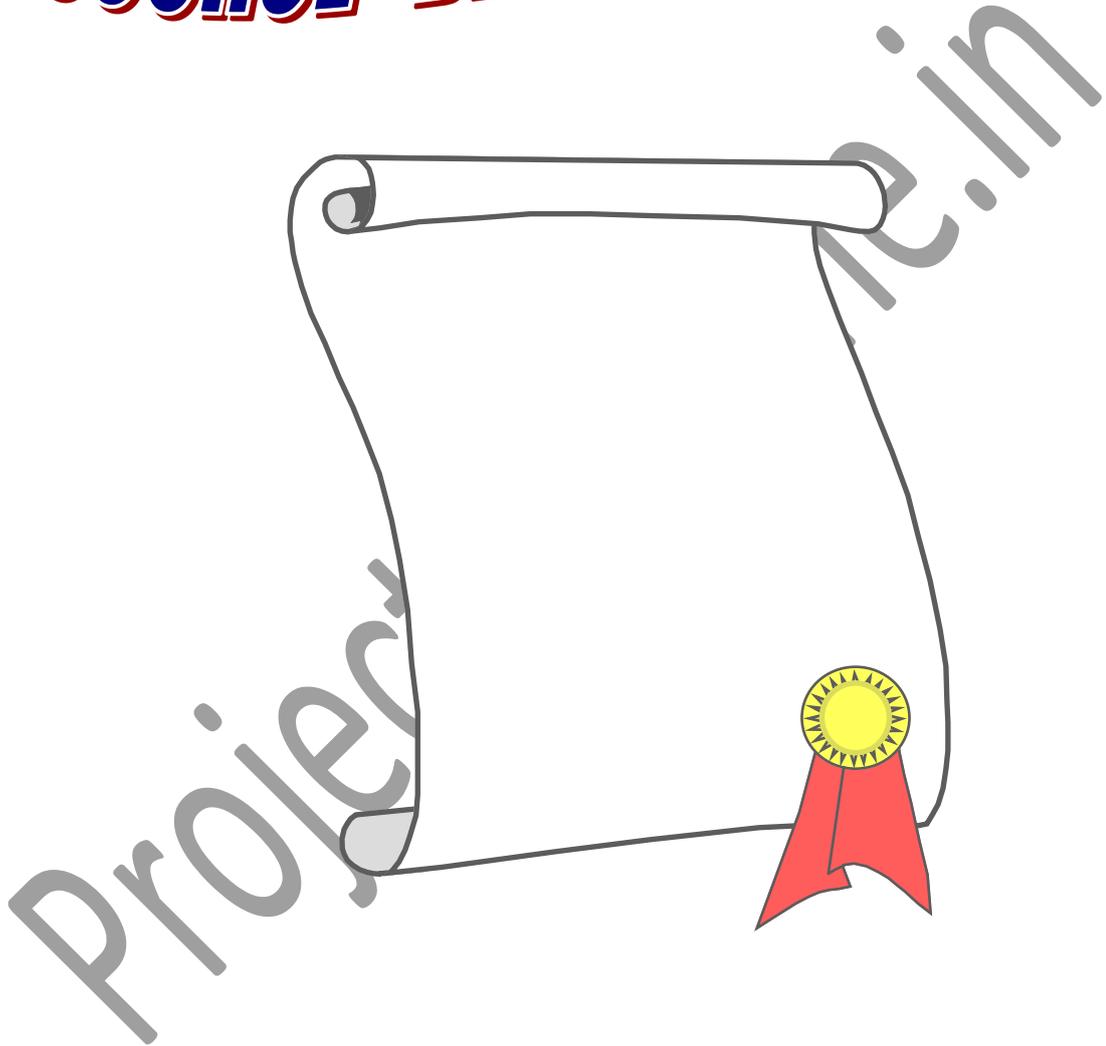
Interdependencies among tasks may be defined using a task network. Tasks, sometimes called the project Work Breakdown Structure (WBS) are defined for the product as a whole or for individual functions.

Both PERT and CPM provide quantitative tools that allow the software planner to (1) determine the critical path-the chain of tasks that determines the duration of the project; (2) establish "most likely" time estimates for individual tasks by applying statistical models; and (3) calculate "boundary times" that define a time window" for a particular task.

Boundary time calculations can be very useful in software project scheduling. Slippage in the design of one function, for example, can retard further development of other functions. It describes important boundary times that may be discerned from a PERT or CPM network: (1) the earliest time that a task can begin when preceding tasks are completed in the shortest possible time, (2) the latest time for task initiation before the minimum project completion time is delayed, (3) the earliest finish-the sum of the earliest start and the task duration, (4) the latest finish- the latest start time added to task duration, and (5) the total float-the amount of surplus time or leeway allowed in scheduling tasks so that the network critical path maintained on schedule. Boundary time calculations lead to a determination of critical path and provide the manager with a quantitative method for evaluating progress as tasks are completed.

Both PERT and CPM have been implemented in a wide variety of automated tools that are available for the personal computer. Such tools are easy to use and take the scheduling methods described previously available to every software project manager.

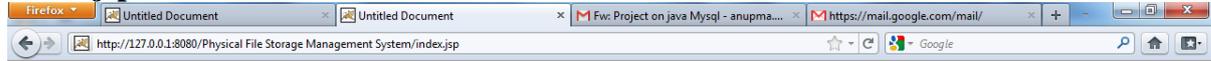
SOURCE DEVELOPMENT



5. SOURCE

5.1 Source Code

Index.jsp



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [CUSTOMER LOGIN](#) [EMPLOYEE LOGIN](#) [ADMIN LOGIN](#)

Welcome to Physical File Storage Management System

Physical File Storage Management System is an online web based system which is responsible for handling the management of Customers's file. In Physical File Storage Management System a customer and an employee of staff is being created by the administration. The Employee can be register itself by new employee signup form. All the files are created by employee after his login and File status is being issued to employee by the administrator of the company and after work of file employee can update the status of file is completed. The Employee can check his file issued history and can view his profile also. When file work will be completed and employee will update status of file by login. Physical File Storage Management System is primarily meant for giving its services to its customers it gives full freedom to check the track of his files and check total files in storage. In PFSMS a customer can be created by administrator and a customer can be register himself by customer sign up form. Customers can also view their profile also. This project will automate all files status i.e. which file is created by which employee and for which customer, what is status (assigned or completed) and File detail (file name, file type, file location, file date, status). Physical File Storage Management System, administrator can check the customers report, employees report, new created files details, issued to employee file reports, Beside that administrator can issue file to any employee and it will update the file status is Assigned.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
background-color: #FFFFFF;
}
.style1 {
font-size: 24px;
font-weight: bold;
}
-->
</style></head>
```

```

<body>
<center>

<table width="800" height="336" border="0" align="center">
  <tr>
    <td height="65" ><% @ include file="header.jsp" %> </td>
  </tr>
  <tr>

    <td width="900" align="left"><label>
      </label>
      <p align="center" class="style1" style="margin-left:20px">Welcome to Physical File
      Storage Management System
      <p >Physical File Storage Management System is an online web based system which is
      responsible for handling the management of Customers's file. In Physical File Storage
      Management System a customer and an employee of staff is being created by the
      administration.The Employee can be register itself by new employee signup form. All the
      files are created by employee after his login and File status is being issued to employee by
      the administrator of the company and after work of file employee can update the status of file
      is completed. The Employee can check his file issued history and can view his profile
      also.When file work will be completed and employee will update status of file by login.

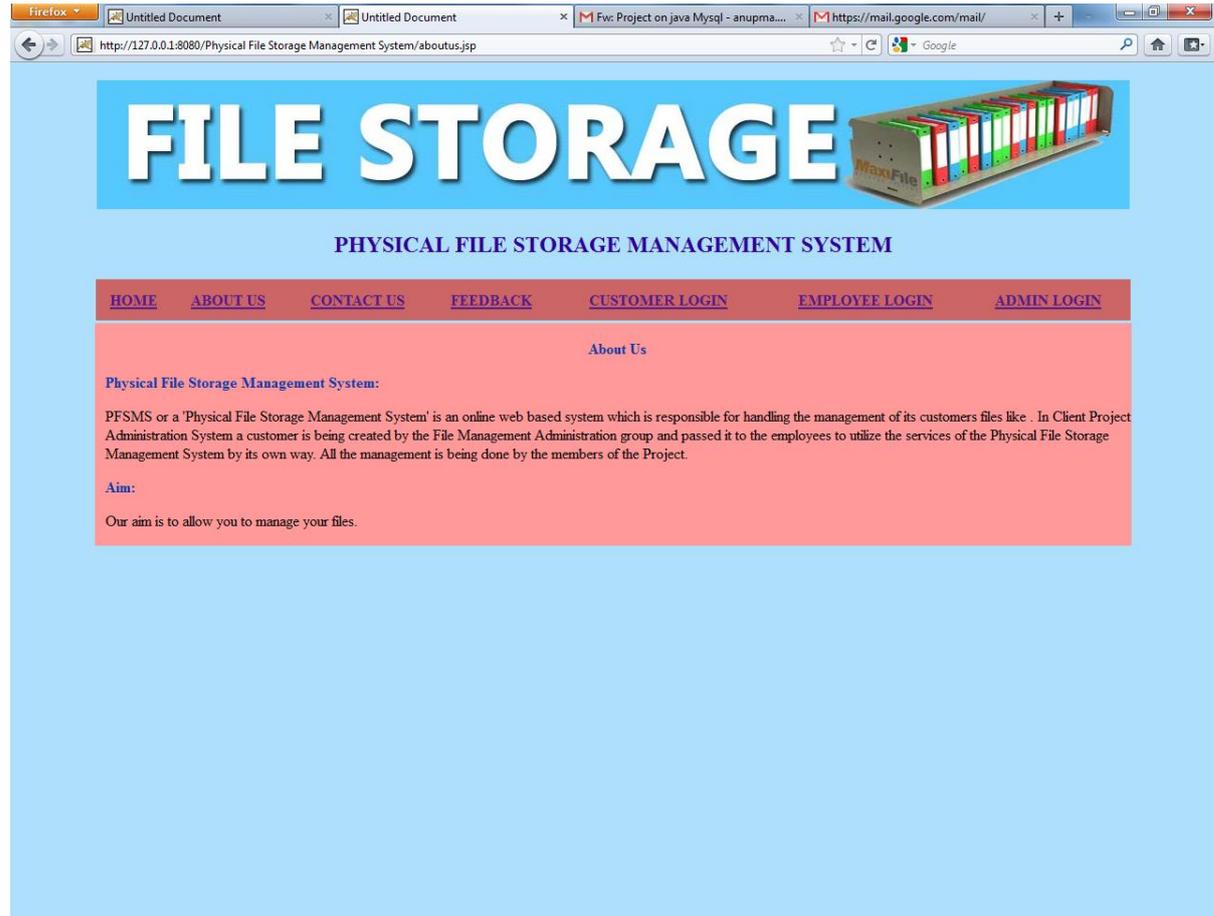
Physical File Storage Management System is primarily meant for giving its services to its
customers it gives full freedom to check the track of his files and check total files in
storage.In PFSMS a customer can be created by administrator and a customer can be register
himself by customer sign up form. Customers can also view their profile also.This project
will automate all files status i.e. which file is created by which employee and for which
customer, what is status (assigned or completed) and File detail (file name, file type, file
location, file date, status).

Physical File Storage Management System , administrator can check the customers report,
employees report,new created files details,issued to employee file reports, Beside that
administrator can issue file to any employee and it will update the file status is Assigned.
</td>
  </tr>
</table>
<div align="center"></div>

</center>
</body>
</html>

```

About us



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
margin-right: 0px;
margin-bottom: 0px;
}
.style1 {
color: #0C3CAC;
font-weight: bold;
margin-left: 10px;
}
-->
</style>
</head>

<body>
```

```

<script language="javascript">
function validate(feedback)
{
if(feedback.textfield.value==0)
{
alert("Enter Name");
feedback.textfield.focus();
return false;
}
if(feedback.textfield2.value==0)
{
alert("Enter E-mail ID");
feedback.textfield2.focus();
return false;
}
if(feedback.textfield3.value==0)
{
alert("Enter Phone no.");
feedback.textfield3.focus();
return false;
}
if(feedback.textfield4.value==0)
{
alert("Enter State");
feedback.textfield4.focus();
return false;
}
if(feedback.textfield5.value==0)
{
alert("Enter Comment");
feedback.textfield5.focus();
return false;
}
}
}
</script>
<form id="feedback" name="feedback" method="post" action="feedback_action.jsp"
onsubmit="return validate(this)">
<table width="1091" height="403" align="center">
<tr>
<td height="65" colspan="2"> <% @ include file="header.jsp" %></td>
</tr>
<tr>
<td bgcolor="#FF9999">

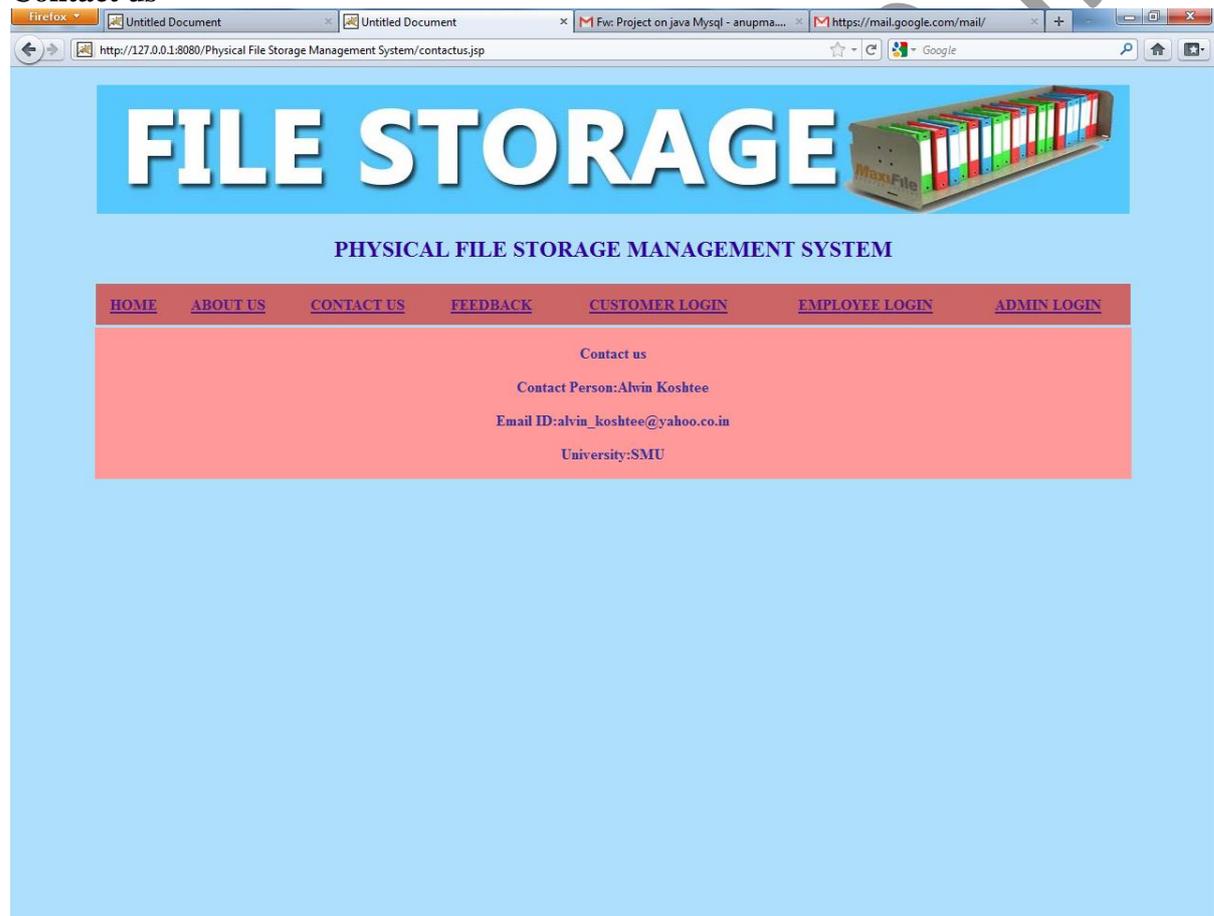
<div align="center">
<p align="center" class="style1">About Us</p>
<p align="left" class="style1">Physical File Storage Management System: </p>
<p align="left" style="margin-left:10px;">PFSMS or a 'Physical File Storage
Management System' is an online web based system which is responsible for handling the
management of its customers files like . In Physical File Storage Management System a

```

customer is being created by the File Management Administration group and passed it to the employees to utilize the services of the Physical File Storage Management System by its own way. All the management is being done by the members of the Project.</p>

```
<p align="left" class="style1">Aim:</p>
<p align="left" style="margin-left:10px;">Our aim is to allow you to manage your
files. </p>
</div></td>
</tr>
</table>
</form>
</body>
</html>
```

Contact us



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
```

```

        margin-left: 0px;
        margin-top: 0px;
        margin-right: 0px;
        margin-bottom: 0px;
    }
    .style1 {color: #333399}
-->
</style>
</head>

<body>
<script language="javascript">
function validate(feedback)
{
if(feedback.textfield.value==0)
{
alert("Enter Name");
feedback.textfield.focus();
return false;
}
if(feedback.textfield2.value==0)
{
alert("Enter E-mail ID");
feedback.textfield2.focus();
return false;
}
if(feedback.textfield3.value==0)
{
alert("Enter Phone no.");
feedback.textfield3.focus();
return false;
}
if(feedback.textfield4.value==0)
{
alert("Enter State");
feedback.textfield4.focus();
return false;
}
if(feedback.textfield5.value==0)
{
alert("Enter Comment");
feedback.textfield5.focus();
return false;
}
}
}
</script>
<table width="1091" height="403" align="center">
<tr>
<td height="65" colspan="2"> <% @ include file="header.jsp" %></td>
</tr>

```

```

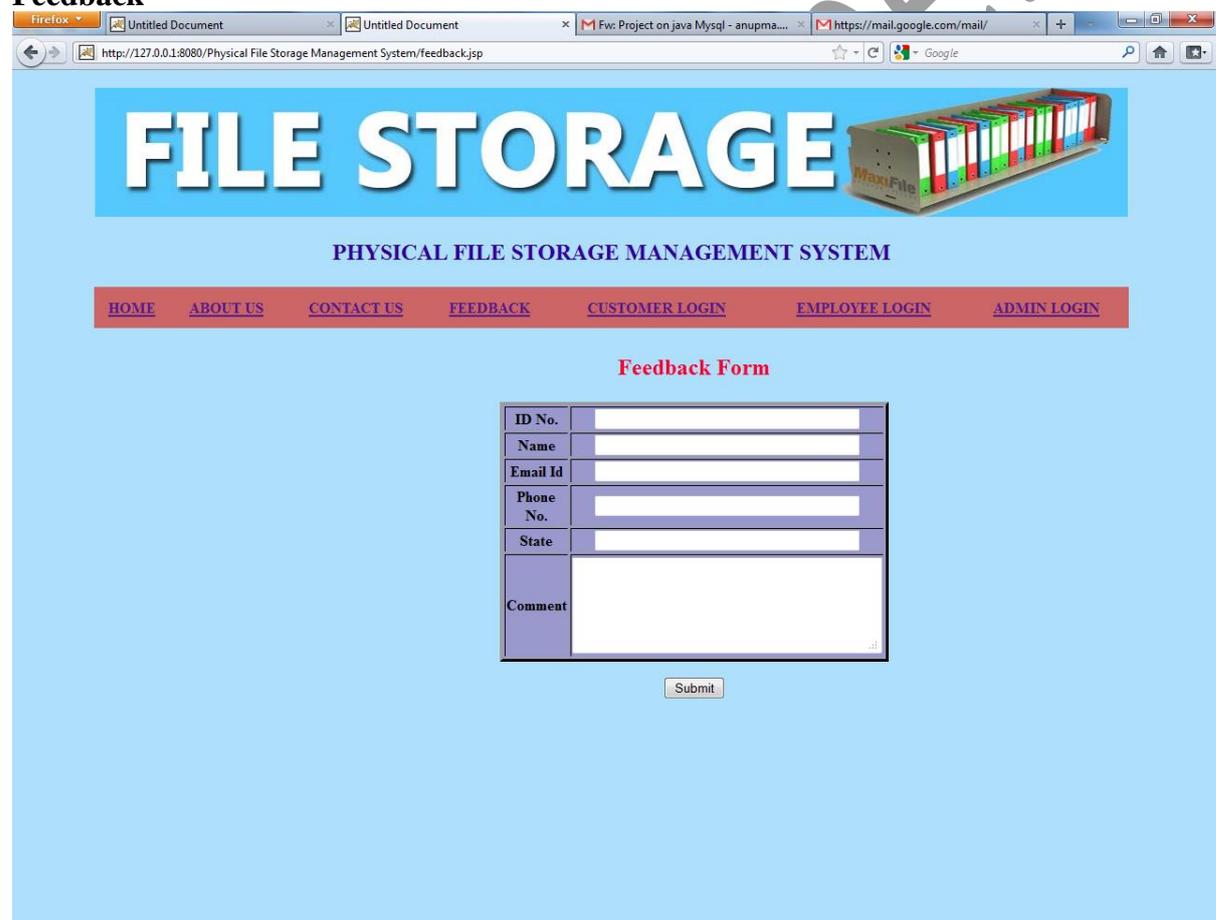
<tr>
  <td bgcolor="#FF9999">

    <div align="center">
      <p align="center"><span class="style1"><strong>Contact us</strong></span></p>
      <p align="center" class="style1"><strong>Contact Person:Alwin Koshtee</strong></p>

      <p align="center" class="style1"><strong>Email
ID:alvin_koshtee@yahoo.co.in</strong></p>
      <p align="center" class="style1"><strong>University:SMU</strong></p>
    </div></td>
</tr>
</table>
</body>
</html>

```

Feedback



```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">

```

```
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
.style2 {
    color: #FF0033;
    font-size: 24px;
    font-weight: bold;
}
-->
</style>
</head>

<body>
<script language="javascript">
function validate(feedback)
{
if(feedback.textfield6.value==0)
{
alert("ID No.");
feedback.textfield6.focus();
return false;
}
if(feedback.textfield6.value==0)
{
alert("Enter Name");
feedback.textfield.focus();
return false;
}
if(feedback.textfield2.value==0)
{
alert("Enter E-mail ID");
feedback.textfield2.focus();
return false;
}
if(feedback.textfield3.value==0)
{
alert("Enter Phone no.");
feedback.textfield3.focus();
return false;
}
if(feedback.textfield4.value==0)
{
alert("Enter State");
feedback.textfield4.focus();
return false;
}
if(feedback.textfield5.value==0)
{
```

```

alert("Enter Comment");
feedback.textfield5.focus();
return false;
}
}
</script>
<form id="feedback" name="feedback" method="post" action="feedback_action.jsp"
onsubmit="return validate(this)">
<table width="1091" height="403" align="center">
  <tr>
    <td height="65" colspan="2"> <% @ include file="header.jsp" %> </td>
  </tr>
  <tr>
    <td width="169">
      </td>
    <td width="906">

      <p align="center" class="style2">Feedback Form </p>
      <table width="372" height="275" border="3" align="center" bordercolor="#000000"
bgcolor="#9999CC">
        <tr>
          <td><div align="center"><strong>ID No.</strong></div></td>
          <td><div align="center">
            <label>
              <input name="textfield6" type="text" size="42" />
            </label>
          </div></td>
        </tr>
        <tr>
          <td><div align="center"><strong>Name</strong></div></td>
          <td><div align="center">
            <label>
              <input name="textfield" type="text" size="42" />
            </label>
          </div></td>
        </tr>
        <tr>
          <td><div align="center"> <strong>Email Id </strong></div></td>
          <td><div align="center">
            <label>
              <input name="textfield2" type="text" size="42" />
            </label>
          </div></td>
        </tr>
        <tr>
          <td><div align="center"><strong>Phone No.</strong></div></td>
          <td><div align="center">
            <label>
              <input name="textfield3" type="text" size="42" />
            </label>
          </div>

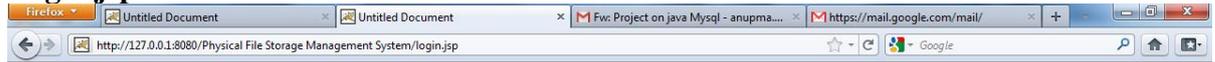
```

```

        </div></td>
</tr>
<tr>
  <td><div align="center"><strong>State</strong></div></td>
  <td><div align="center">
    <label>
      <input name="textfield4" type="text" size="42" />
    </label>
  </div></td>
</tr>
<tr>
  <td height="95"><div align="center">
    <div align="center"><strong>Comment</strong></div>
  </div></td>
  <td><div align="center">
    <label>
      <textarea name="textfield5" cols="38" rows="5"></textarea>
    </label>
  </div></td>
</tr>
</table>
<p align="center">
  <label>
    <input type="submit" name="Submit" value="Submit" />
  </label>
</p>
<label>
  </label>
<p align="center">&nbsp;</p></td>
</tr>
</table>
</form>
</body>
</html>

```

Login.jsp



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [CUSTOMER LOGIN](#) [EMPLOYEE LOGIN](#) [ADMIN LOGIN](#)

CUSTOMER LOGIN FORM

Username

Password

[New User Sign Up](#)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    background-color: #FFFFFF;
}
-->
</style></head>

<body>
<script language="javascript">
function validate(login)
{
if(login.textfield.value=="")
{
```

```

alert("Enter Username");
login.textfield.focus();
return false;
}
if(login.textfield2.value=="")
{
alert("Enter Emp. ID");
login.textfield2.focus();
return false;
}
if(login.textfield3.value=="")
{
alert("Enter Email ID");
login.textfield3.focus();
return false;
}
if(login.textfield4.value=="")
{
alert("Enter Password");
login.textfield4.focus();
return false;
}
}
}
</script>
<center>
<form id="login" name="login" method="post" action="login_action.jsp" onsubmit="return
validate(this)">
<table width="1091" height="336" border="0">
<tr>
<td height="65" colspan="2"> <% @ include file="header.jsp" %></td>
</tr>
<tr>
<td width="168" align="left" valign="top"></td>
<td width="900"><table width="400" height="192" align="center" style="border: 3px
solid #941c03;" bgcolor="#f9ebba">
<tr>
<td colspan="2"><div align="center" style="color:#941c03"><strong>CUSTOMER
LOGIN FORM </strong></div></td>
</tr>
<tr>
<td width="82" height="35"><div
align="center"><strong>Username</strong></div></td>
<td width="216">
<label></label>
<div align="center">
<label>
<input name="uname" type="text" size="40" />
</label>
</div>
</td>

```

```

</tr>

<tr>
<td><div align="center"><strong>Password</strong></div></td>
<td><div align="center">

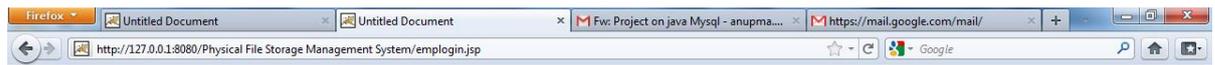
<label>
<input name="pswd" type="password" size="40" />
</label>
</div></td>
</tr>
<tr>
<td colspan="2"><div align="center"><strong><input type="submit" name="Submit"
value="Submit" style="background-color:#92181c; border-color:#92181c; color:#FFFFFF"
/>

<a href="newuser.jsp" style="color:#92181c"> New User Sign Up</a>
</div></td>
</tr>
</table>

<p align="center">&nbsp;</p></td>
</tr>
</table>
<div align="center"></div>
</form>
</center>
</body>
</html>

```

Emplogin.jsp



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [CUSTOMER LOGIN](#) [EMPLOYEE LOGIN](#) [ADMIN LOGIN](#)

EMPLOYEE LOGIN FORM

Username

Password

[New Client Sign Up](#)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    background-color: #FFFFFF;
}
-->
</style></head>

<body>
<script language="javascript">
function validate(login)
{
if(login.textfield.value=="")
```

```

{
alert("Enter Username");
login.textfield.focus();
return false;
}
if(login.textfield2.value=="")
{
alert("Enter Emp. ID");
login.textfield2.focus();
return false;
}
if(login.textfield3.value=="")
{
alert("Enter Email ID");
login.textfield3.focus();
return false;
}
if(login.textfield4.value=="")
{
alert("Enter Password");
login.textfield4.focus();
return false;
}
}
</script>
<center>
<form id="login" name="login" method="post" action="emplogin_action.jsp"
onsubmit="return validate(this)">
<table width="1091" height="336" border="0">
<tr>
<td height="65" colspan="2"><%@ include file="header.jsp" %></td>
</tr>
<tr>
<td width="168" align="left" valign="top"></td>
<td width="900"><table width="329" height="192" style="border: 3px solid #003366;"
align="center" bgcolor="#79CDDA">
<tr>
<td colspan="2"><div align="center" style="color:#003366"><strong>EMPLOYEE
LOGIN FORM </strong></div></td>
</tr>
<tr>
<td width="82" height="35"><div
align="center"><strong>Username</strong></div></td>
<td width="216">
<label></label>
<div align="center">
<label>
<input name="uname" type="text" id="uname" size="40" />
</label>
</div>

```

```

        </td>
    </tr>

    <tr>
        <td><div align="center"><strong>Password</strong></div></td>
        <td><div align="center">

            <label>
                <input name="pswd" type="password" id="pswd" size="40" />
            </label>
        </div></td>
    </tr>

    <tr>
        <td colspan="2"><div align="center"><strong><input type="submit" name="Submit"
value="Submit" style="background-color:#003366; border-color:#003366; color:#FFFFFF"
/>

        <a href="newclient.jsp" style="color:#003366"> New Client Sign Up</a>
        </div></td>
    </tr>
</table>

<label>

    <p align="center">&nbsp;</p></td>
</tr>
</table>
<div align="center"></div>
</form>
</center>
</body>
</html>

```

admin_login_form.jsp

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}

```

```

.style5 {
    font-size: 24px;
    color: #FF0033;
}
-->
</style></head>

<body>
<script language="javascript">
function validate(adminlogin)
{
if(adminlogin.textfield.value=="")
{
alert("Enter Username");
adminlogin.textfield.focus();
return false;
}
if(adminlogin.textfield2.value=="")
{
alert("Enter Password");
adminlogin.textfield2.focus();
return false;
}
}
</script>
<center>
<form id="adminlogin" name="adminlogin" method="POST" action="admin_action.jsp"
onsubmit="return validate(this)">
<table width="1091" height="336" align="center">
<tr>
<td height="65" colspan="2"><%@ include file="header.jsp" %></td>
</tr>
<tr>
<td width="168" valign="top">&nbsp;</td>
<td width="900"><table width="400" height="192" align="center" style="border: 3px
solid #941c03;" bordercolor="#e5494c" bgcolor="#D4D4D4">
<tr>
<td colspan="2"><div align="center" style="color:#941c03"><strong>Administrator
Login Form </strong></div></td>
</tr>
<tr>
<td width="82" height="35"><div
align="center"><strong>Username</strong></div></td>
<td width="216">
<label></label>
<div align="center">
<label>
<input name="textfield" type="text" size="40" />
</label>
</div>

```

```

        </td>
    </tr>

    <tr>
        <td><div align="center"><strong>Password</strong></div></td>
        <td><div align="center">

            <label>
                <input name="textfield2" type="password" size="40" />
            </label>
        </div></td>
    </tr>
    <tr>
        <td colspan="2"><div align="center"><strong><input type="submit" name="Submit"
value="Submit" style="background-color:#92181c; border-color:#92181c; color:#FFFFFF"
/></strong>
        </div></td>
    </tr>
</table>
</td>
</tr>
</table>
</form>
</center>
</body>
</html>

```

Projecthelpline.in

admin_menu.jsp

FILE STORAGE

PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [CUSTOMER LOGIN](#) [EMPLOYEE LOGIN](#) [LOGOUT](#)

Administrator Menu

[Admin DashBoard](#)

[Add Customer Details](#)

[Add Employee Details](#)

[Issue File to Employee](#)

[Report Available](#)

[Employee Details](#)

[Client Details](#)

[Files in Storage](#)

[Files Assigned Report](#)

[Feedback Details](#)

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
.style5 {color: #FF0033; font-size: 24px;}
a:link {
    color: #99FFFF;
}
a:hover {
    color: #0066FF;
}
-->
</style></head>
```

```

<body>
<table width="1091" height="336" align="center">
  <tr>
    <td height="65" colspan="2"> <% @ include file="header.jsp" %></td>
  </tr>
  <tr>
    <td width="168" valign="top"><p>&nbsp;</p>
      <% @ include file="menu.jsp" %>
    </td>
    <td width="900">
      <p align="center" class="style5">Administrator Menu</p>
      <table width="201" height="199" border="3" align="center" cellpadding="7"
        cellspacing="7" bordercolor="#000000" bgcolor="#CC6666">
        <tr>
          <td><div align="center"><strong>Report Available </strong></div></td>
        </tr>
        <tr>
          <td><div align="center"><a href="show_emp_detail.jsp">Employee
            Details</a></div></td>
        </tr>
        <tr>
          <td><div align="center"><a href="show_cust_detail.jsp">Client
            Details</a></div></td>
        </tr>
        <tr>
          <td><div align="center"><a href="total_files.jsp">Files in Storage</a></div></td>
        </tr>
        <tr>
          <td><div align="center"><a href="show_file_issue.jsp">Files Assigned
            Report</a></div></td>
        </tr>
        <tr>
          <td><div align="center"><a href="show_feedback.jsp">Feedback
            Details</a></div></td>
        </tr>
      </table>
    </td>
  </tr>
</table>
</body>
</html>

```

add_cust.jsp

FILE STORAGE

PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

HOME ABOUT US CONTACT US FEEDBACK CUSTOMER LOGIN EMPLOYEE LOGIN LOGOUT

Admin Menu

Admin DashBoard
Add Customer Details
Add Employee Details
Issue File to Employee

CUSTOMER DETAILS

Customer Username	<input type="text"/>
Password	<input type="password"/>
Customer Name	<input type="text"/>
Company Name	<input type="text"/>
Address	<input type="text"/>
City	<input type="text"/>
State	<input type="text"/>
Email	<input type="text"/>
Phone	<input type="text"/>

Submit

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
.style2 {
    font-size: 24px;
    color: #FF0033;
}
-->
</style><script language="javascript">
function validate(empdetail)
{
if(emp_detail.textfield.value=="")
{
```

```
alert("Enter Client ID");
emp_detail.textfield.focus();
return false;
}
if(emp_detail.textfield2.value=="")
{
alert("Enter Company Name");
emp_detail.textfield2.focus();
return false;
}
if(emp_detail.textfield8.value=="")
{
alert("Enter Website");
emp_detail.textfield8.focus();
return false;
}
if(emp_detail.textfield7.value=="")
{
alert("Enter City");
emp_detail.textfield7.focus();
return false;
}
if(emp_detail.textfield4.value=="")
{
alert("Enter State");
emp_detail.textfield4.focus();
return false;
}
if(emp_detail.textfield5.value=="")
{
alert("Enter Contact Person");
emp_detail.textfield5.focus();
return false;
}
if(emp_detail.textfield6.value=="")
{
alert("Enter Email ID");
emp_detail.textfield6.focus();
return false;
}
}
}
</script></head>
```

<body>

<table width="1091" height="336" align="center">

<tr>


```

        <td><div align="center">Address</div></td>
        <td><label>
            <input name="addr" type="text" id="addr" size="30" />
        </label></td>
    </tr>
    <tr>
        <td><div align="center">City</div></td>
        <td><input name="city" type="text" id="city" size="30" /></td>
    </tr>
    <tr>
        <td><div align="center">State</div></td>
        <td><input name="state" type="text" id="state" size="30" /></td>
    </tr>
    <tr>
        <td><div align="center">Email</div></td>
        <td><input name="email" type="text" id="email" size="30" /></td>
    </tr>
    <tr>
        <td><div align="center">Phone</div></td>
        <td><input name="phn" type="text" id="phn" size="30" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>
<p align="center">
    <label>
        <input type="submit" name="Submit" value="Submit" />
    </label>
</p>
</form>
</td></tr>
</table>
</body>
</html>

```

show_cust_detail.jsp

FILE STORAGE

PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [CUSTOMER LOGIN](#) [EMPLOYEE LOGIN](#) [LOGOUT](#)

[Admin DashBoard](#)
[Add Customer Details](#)
[Add Employee Details](#)
[Issue File to Employee](#)

[Admin Menu](#)

Customer Details

Customer ID	Username	Password	Name	Company Name	Address	City	State	Email ID	Phone No.
2	Vikas1	123	Vikas Narang	infotech pvt. ltd.	Preet Vihar	Delhi	New Delhi	vicky@gmail.com	9876543212

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
.style6 {
    font-size: 24px;
    color: #FF0033;
}
table tr td{font-size:16px;}
-->
</style></head>

<body>
```

```

<table width="1091" height="336" align="center">
  <tr>
    <td height="65" colspan="2"> <p>&nbsp;</p>
    <% @ include file="header.jsp" %></td>
  </tr>
  <tr>
    <td width="168" valign="top"><p>&nbsp;</p>
    <% @ include file="menu.jsp" %></td>
    <td width="900">
      <p align="center"><a href="admin_menu.jsp">Admin Menu</a></p>
      <p align="center" class="style6">Customer Details </p>
      <table width="700" height="89" border="3" align="center" bordercolor="#000000"
      bgcolor="#9999CC">
        <tr>
          <td width="81"><div align="center"><strong>Customer
          ID</strong></div></td>
          <td width="92"><div
          align="center"><strong>Username</strong></div></td>
          <td width="92"><div
          align="center"><strong>Password</strong></div></td>
          <td width="92"><div align="center"><strong>Name</strong></div></td>
          <td width="92"><div align="center"><strong>Company
          Name</strong></div></td>
          <td width="92"><div align="center"><strong>Address</strong></div></td>
          <td width="92"><div
          align="center"><strong>City</strong></div></td>
          <td width="92"><div align="center"><strong>State</strong></div></td>
          <td width="100"><div align="center"><strong>Email
          ID</strong></div></td>
          <td width="92"><div align="center"><strong>Phone
          No.</strong></div></td>
        </tr>
      </table>
    </td>
  </tr>
<% @ include file="connect.jsp" %> <%
  ResultSet rs1=st.executeQuery("select * from customer");
  while(rs1.next())
  {
    %>
    <tr>
      <td><center><%=rs1.getString("id")%> </center></td>
      <td><center><%=rs1.getString("username")%></center></td>
      <td><center><%=rs1.getString("password")%> </center></td>
      <td><center><%=rs1.getString("name")%> </center></td>
      <td><center><%=rs1.getString("cname")%></center></td>
      <td><center><%=rs1.getString("address")%> </center></td>
    </tr>
  }
  %>
</pre>

```

```
<td><center><%=rs1.getString("city")%> </center></td>
<td><center><%=rs1.getString("state")%> </center></td>
<td><center><%=rs1.getString("email")%> </center></td>
<td><center><%=rs1.getString("phone")%></center></td>
```

```
</tr>
<%
}
%>
</table>
</td>
</tr>
</table>
</body>
</html>
```

Projecthelpline.in

add_emp.jsp

The screenshot shows a web browser window with the following elements:

- Browser tabs: Firefox, Untitled Document, Fw: Project on java Mysql, https://mail.g...gle.com/mail/
- Address bar: http://127.0.0.1:8080/Physical File Storage Management System/add_emp.jsp
- Header: **FILE STORAGE** with a file cabinet image.
- Sub-header: **PHYSICAL FILE STORAGE MANAGEMENT SYSTEM**
- Navigation bar: HOME, ABOUT US, CONTACT US, FEEDBACK, CUSTOMER LOGIN, EMPLOYEE LOGIN, LOGOUT
- Main content: **Admin Menu** and **EMPLOYEE DETAILS**
- Left sidebar menu: Admin DashBoard, Add Customer Details, Add Employee Details, Issue File to Employee
- Form fields: Employee ID, Employee Username, Password, Employee Name, Designation, Department, Email Id, Phone
- Submit button

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
}
.style2 {
    font-size: 24px;
    color: #FF0033;
}
-->
</style><script language="javascript">
function validate(empdetail)
{
if(emp_detail.textfield.value=="")
{
```

```
alert("Enter Client ID");
emp_detail.textfield.focus();
return false;
}
if(emp_detail.textfield2.value=="")
{
alert("Enter Company Name");
emp_detail.textfield2.focus();
return false;
}
if(emp_detail.textfield8.value=="")
{
alert("Enter Website");
emp_detail.textfield8.focus();
return false;
}
if(emp_detail.textfield7.value=="")
{
alert("Enter City");
emp_detail.textfield7.focus();
return false;
}
if(emp_detail.textfield4.value=="")
{
alert("Enter State");
emp_detail.textfield4.focus();
return false;
}
if(emp_detail.textfield5.value=="")
{
alert("Enter Contact Person");
emp_detail.textfield5.focus();
return false;
}
if(emp_detail.textfield6.value=="")
{
alert("Enter Email ID");
emp_detail.textfield6.focus();
return false;
}
}
}
</script></head>
```

<body>

<table width="1091" height="336" align="center">

<tr>


```

        <td><label>
            <input name="desg" type="text" id="desg" size="30" />
        </label></td>
    </tr>
    <tr>
        <td><div align="center">Department</div></td>
        <td><label>
            <input name="dep" type="text" id="dep" size="30" />
        </label></td>
    </tr>
    <tr>
        <td><div align="center">Email Id </div></td>
        <td><input name="email" type="text" id="email" size="30" /></td>
    </tr>
    <tr>
        <td><div align="center">Phone</div></td>
        <td><input name="phn" type="text" id="phn" size="30" /></td>
    </tr>
    <tr>
        <td>&nbsp;</td>
        <td>&nbsp;</td>
    </tr>
</table>
<p align="center">
    <label>
        <input type="submit" name="Submit" value="Submit" />
    </label>
</p>

</form>
</td></tr>
</table>

</body>
</html>

```



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [CUSTOMER LOGIN](#) [LOGOUT](#) [ADMIN LOGIN](#)

- [Your Profile](#)
- [Add new File](#)
- [Your Assigned File](#)
- [Update File Status](#)

Welcome Anupma123

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
background-color: #FFFFFF;
}
.style1 {
color: #FF0033;
font-size: 24px;
}
-->
</style></head>

<body>
<center>

<table width="1091" height="336" border="0">
```

```

<tr>
  <td height="65" colspan="2"> <% @ include file="header.jsp"%>
</td>
</tr>
<tr>
  <td width="168" align="left" valign="top"><p>&nbsp;</p>
    <% @ include file="empMenu.jsp"%>

  </td>
  <td width="900"><label>
    </label>

    <p align="center">
      <%

        String u=(String)session.getAttribute("empname");
        if(u != null)
        {
          out.println("Welcome "+ u);
        }
        else
        {
          response.sendRedirect("emplogin.jsp");
        }
      %>
    </td>
  </tr>
</table>
<div align="center"></div>

</center>
</body>
</html>

```

Projecthelp.in

FILE STORAGE

PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

- [HOME](#)
- [ABOUT US](#)
- [CONTACT US](#)
- [FEEDBACK](#)
- [CUSTOMER LOGIN](#)
- [LOGOUT](#)
- [ADMIN LOGIN](#)

Welcome Anupma123

- [Your Profile](#)
- [Add new File](#)
- [Your Assigned File](#)
- [Update File Status](#)

Add new File

File Number	<input type="text"/>
File Name	<input type="text"/>
File Type	<input type="text"/>
File Submitted Date	<input type="text"/>
File Location	Head Office
Select Customer Id	2
Description	<input type="text"/>

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
background-color: #FFFFFF;
}
-->
</style></head>

<body>
<% @ include file="connect.jsp" %>

<center>
```

```

<table width="1091" height="336" border="0">
  <tr>
    <td height="65" colspan="2"> <%@ include file="header.jsp" %></td>
  </tr>
  <tr>
    <td width="168" align="left" valign="top"><p>
      <%@ include file="empMenu.jsp" %>

      </p>
    </td>
    <td width="900">
<div align="right" style="font-size:18px; color:#000066; font-weight:bold"> <%

      String un=(String)session.getAttribute("empname");
      if(un != null)
      {
        out.println("Welcome "+ un);
      }
      else
      {
        response.sendRedirect("emplogin.jsp");
      }
    %>

</div>
    <form action="newFile_action.jsp" method="post">
      <table width="500" height="192" align="center" style="border: 3px solid #941c03;
font-size:14px" bgcolor="#f9ebba">
        <tr>
          <td colspan="2"><div align="center" style="color:#941c03; font-
size:18px;"><strong>Add new File </strong></div></td>
        </tr>
        <tr >
          <td><div align="center" style="font-size:16px"><strong>File Number
</strong></div></td>
          <td width="216">
            <label></label>
            <div align="center">
              <label>
                <input name="fnum" type="text" id="fnum" size="40" />
              </label>
            </div>
          </td>
        </tr>
        <tr >
          <td><div align="center" style="font-size:16px"><strong>File Name
</strong></div></td>
          <td width="216">
            <label></label>
            <div align="center">
              <label>

```

```

        <input name="fname" type="text" id="fname" size="40" />
    </label>
</div>
</td>
</tr>
<tr>
    <td><div align="center" style="font-size:16px"><strong> File Type
</strong></div></td>
    <td><div align="center">

        <label>
            <input name="ftype" type="text" id="ftype" size="40" />
        </label>

        </div></td>
</tr>
<tr>
    <td><div align="center" style="font-size:16px"><strong> File Submitted Date
</strong></div></td>
    <td><div align="center">

        <label>
            <input name="fdate" type="text" id="fdate" size="40" />
        </label>

        </div></td>
</tr>
<tr>
    <td><div align="center" style="font-size:16px"><strong> File Location
</strong></div></td>
    <td><div align="center">

        <label>
            <select name="flocation" style="width:270px">
                <option value="Head Office">Head Office</option>
                <option value="Branch 1">Branch 1</option>
                <option value="Branch 2">Branch 2</option>
                <option value="Branch 3">Branch 3</option>
                <option value="Branch 4">Branch 4</option>
            </select>

        </label>

        </div></td>
</tr>
<tr>
    <td><div align="center" style="font-size:16px"><strong>Select Customer Id
</strong></div></td>

```

```

<td><div align="center">

    <label>
    <select name="cust" style="width:270px">
        <%
            ResultSet rsc=st.executeQuery("select * from customer");
while(rsc.next())
{
            %>

                <option value="<%= rsc.getString("id") %>"><%= rsc.getString("id")
%></option>

                <%
                }
                %>
            </select>

        </label>

    </div></td>
</tr>

    <tr>
    <td align="center" style="font-size:16px"><strong>Description</strong></td>
    <td align="center">

        <label>

            <textarea name="desc" cols="30" rows="5"> </textarea>

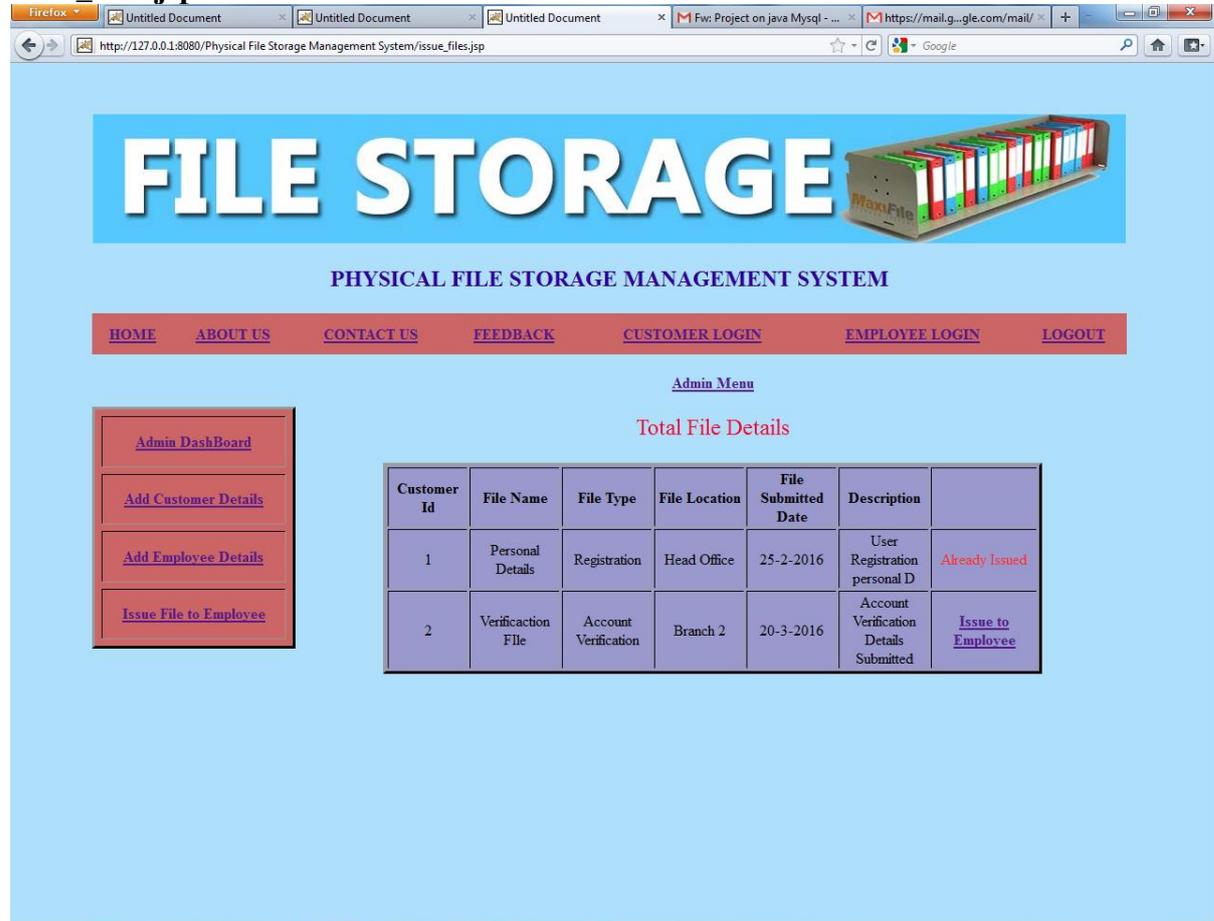
        </label>
    </td>
</tr>
    <tr>
    <td colspan="2"><div align="center"><strong><input type="submit" name="Submit"
value="Add Detail" style="background-color:#92181c; border-color:#92181c;
color:#FFFFFF" />
</strong>

    </div></td>
</tr>
</table>
</form>
</td>
</tr>
</table>
</center>

```

```
</body>
</html>
```

issue_files.jsp



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
}
.style6 {
font-size: 24px;
color: #FF0033;
}
table tr td{font-size:16px;}

```

```

-->
</style></head>

<body>

<table width="1091" height="336" align="center">
  <tr>
    <td height="65" colspan="2"> <p>&nbsp;</p>
    <% @ include file="header.jsp" %></td>
  </tr>
  <tr>
    <td width="168" valign="top"><p>&nbsp;</p>
    <% @ include file="menu.jsp" %></td>
    <td width="900">
      <p align="center"><a href="admin_menu.jsp">Admin Menu</a></p>
      <p align="center" class="style6">Total File Details </p>
      <table width="700" height="89" border="3" align="center" bordercolor="#000000"
      bgcolor="#9999CC" style="font-size:12px;">
        <tr>
          <td width="81"><div align="center"><strong>Customer
          Id</strong></div></td>
          <td width="92"><div align="center"><strong>File
          Name</strong></div></td>
          <td width="92"><div align="center"><strong>File
          Type</strong></div></td>
          <td width="92"><div align="center"><strong>File
          Location</strong></div></td>
          <td width="92"><div align="center"><strong>File Submitted
          Date</strong></div></td>
          <td width="92"><div
          align="center"><strong>Description</strong></div></td>
          <td></td>
        </tr>
      </table>
    </td>
  </tr>

<% @ include file="connect.jsp" %>
<%
    String id=(String)session.getAttribute("empId");
    ResultSet rs1=st.executeQuery("select * from file_detail");
    while(rs1.next())
    {
    String status=rs1.getString("status");
    %>
    <tr>
    <td><center><%=rs1.getString("customer_id")%> </center></td>
    <td><center><%=rs1.getString("file_name")%></center></td>
    <td><center><%=rs1.getString("file_type")%> </center></td>

```

```

        <td><center><%=rs1.getString("flocation")%> </center></td>
        <td><center><%=rs1.getString("entryDate")%></center></td>
        <td><center><%=rs1.getString("desc")%> </center></td>
        <%
        if(status.equals("0"))
        {
        %>
        <td><center><a href="issue.jsp?id=<%=rs1.getString("id")%>">Issue to
Employee</a> </center></td>
        <%
        }
        else
        {
        %>
        <td style="color:#FF3333"><center>Already Issued </center></td>

        <%
        }
        %>

    </tr>
    <%
    }
    %>
</table>
</td>
</tr>
</table>

</body>
</html>

```

issue.jsp



```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
}
.style2 {
font-size: 24px;
color: #FF0033;
}
-->
</style><script language="javascript">
function validate(empdetail)
{
if(emp_detail.textfield.value=="")
```

```
{
alert("Enter Client ID");
emp_detail.textfield.focus();
return false;
}
if(emp_detail.textfield2.value=="")
{
alert("Enter Company Name");
emp_detail.textfield2.focus();
return false;
}
if(emp_detail.textfield8.value=="")
{
alert("Enter Website");
emp_detail.textfield8.focus();
return false;
}
if(emp_detail.textfield7.value=="")
{
alert("Enter City");
emp_detail.textfield7.focus();
return false;
}
if(emp_detail.textfield4.value=="")
{
alert("Enter State");
emp_detail.textfield4.focus();
return false;
}
if(emp_detail.textfield5.value=="")
{
alert("Enter Contact Person");
emp_detail.textfield5.focus();
return false;
}
if(emp_detail.textfield6.value=="")
{
alert("Enter Email ID");
emp_detail.textfield6.focus();
return false;
}
}
</script></head>
```

<body>

<% @ include file="connect.jsp" %>

<table width="1091" height="336" align="center">

```

<tr>
  <td height="65" colspan="2"> <% @ include file="header.jsp" %></td>
</tr>
<tr>
  <td width="168" valign="top"><p>&nbsp;</p>
  <p>
    <% @ include file="menu.jsp" %>

  </p>
</td>

<td width="900" >

  <p align="center"><a href="admin_menu.jsp">Admin Menu</a></p>
  <p align="center" class="style2">ISSUE FILE</p>
  <%
    String fileId=request.getParameter("id");
    %>
  <form id="emp_detail" name="emp_detail" method="POST" action="issue_action.jsp"
onsubmit="return validate(this)">
<input type="hidden" name="id" value="<%= fileId %>" />
  <table width="600" border="3" align="center" bordercolor="#000000"
bgcolor="#9999CC">

  <tr>
    <td width="135" height="36"><div align="center">Select Employee ID to Issue
File</div></td>
    <td width="187"><label>
      <select name="cust" style="width:270px">
        <%
          ResultSet rem=st.executeQuery("select * from employee");
          while(rem.next())
          {
            <%
              <option value="<%= rem.getString("emp_id") %>"><%=
rem.getString("emp_id") %></option>
            <%
              }
            %>
          </select>
        </label></td>
  </tr>

</table>

```

```

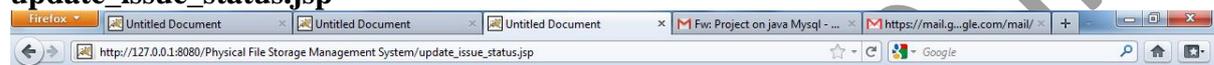
<p align="center">
  <label>
  <input type="submit" name="Submit" value="Submit" />
  </label>
</p>

</form>
</td></tr>
</table>

</body>
</html>

```

update_issue_status.jsp



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

- [HOME](#)
- [ABOUT US](#)
- [CONTACT US](#)
- [FEEDBACK](#)
- [CUSTOMER LOGIN](#)
- [LOGOUT](#)
- [ADMIN LOGIN](#)

Welcome Anupma123

- [Your Profile](#)
- [Add new File](#)
- [Your Assigned File](#)
- [Update File Status](#)

Your Files History

File Number	File Name	File Type	Status
81234	Verification File	Account Verification	Completed

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {

```

```

        margin-left: 0px;
        margin-top: 0px;
        background-color: #FFFFFF;
    }
    .style3 {
        font-size: 24px;
        color: #FF0033;
    }
table tr td{font-size:16px;}

-->
</style></head>

<body>
<center>

<table width="1091" height="336" border="0">
  <tr>
    <td height="65" colspan="2"> <% @ include file="header.jsp"% ></td>
  </tr>
  <tr><td align="right" colspan="2"><div align="right" style="font-size:18px;
color:#000066; font-weight:bold; float:right"><%

        String u=(String)session.getAttribute("empname");
        if(u != null)
        {
            out.println("Welcome " + u);
        }
        else
        {
            response.sendRedirect("emplogin.jsp");
        }
    %>

</div></td></tr>
  <tr>
    <td width="168" align="left" valign="top"><p>
      <% @ include file="empMenu.jsp"% >
    </p>
    </td>
    <td width="900"><p align="center" class="style3">Your Files History</p>
      <table width="700" height="89" border="3" align="center" bordercolor="#000000"
        bgcolor="#9999CC" style="font-size:12px;">
        <tr>
          <td width="92"><div align="center"><strong>File
            Number</strong></div></td>
          <td width="92"><div align="center"><strong>File
            Name</strong></div></td>
          <td width="92"><div align="center"><strong>File
            Type</strong></div></td>

```

```

                <td width="92"><div align="center"><strong>Status</strong></div></td>

        </tr>

<% @ include file="connect.jsp" %>
<%
        String id=(String)session.getAttribute("empId");
        ResultSet rs1=st.executeQuery("select * from issue where empid='"+id+"' ");
        while(rs1.next())
        {
                String status=rs1.getString("status"); %>
                <tr>

                <td><center><%=rs1.getString("fid")%></center></td>
                <td><center><%=rs1.getString("fname")%> </center></td>
                <td><center><%=rs1.getString("ftype")%> </center></td>

                <%
                if(status.equals("Assigned"))
                {
                %>
                <td><center><a
href="issue_update.jsp?issueId=<%=rs1.getString("id")%>&fid=<%=rs1.getString("fid")%>
">Update</a> </center></td>
                <%
                }
                else
                {
                %>
                <td style="color:#FF3333"><center>Completed </center></td>

                <%
                }
                %>

        </tr>
        <%
        }
        %>
</table>

```

```

</td>
</tr>
</table>
</center>
</body>
</html>

```

cliuser.jsp



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#)
[ABOUT US](#)
[CONTACT US](#)
[FEEDBACK](#)
[LOGOUT](#)
[EMPLOYEE LOGIN](#)
[ADMIN LOGIN](#)

Your Profile
Files In Storage
File Track

Welcome Vikas!

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    background-color: #FFFFFF;
}
.style1 {
    color: #FF0033;
    font-size: 24px;
}

```

```

-->
</style></head>

<body>
<center>

<table width="1091" height="336" border="0">
  <tr>
    <td height="65" colspan="2"><% @ include file="header.jsp"%></td>
  </tr>
  <tr>
    <td width="168" align="left" valign="top"><p>&nbsp;</p>
    <% @ include file="user_menu.jsp"%> </td>
    <td width="900"><label>
      </label>

      <p align="center">
        <%

          String u=(String)session.getAttribute("clientname");
          if(u != null)
          {
            out.println("Welcome "+ u);
          }
          else
          {
            response.sendRedirect("login.jsp");
          }
        %>

      </td>
  </tr>
</table>
<div align="center"></div>

</center>
</body>
</html>

```

file_track.jsp



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [LOGOUT](#) [EMPLOYEE LOGIN](#) [ADMIN LOGIN](#)

Your Profile
Files In Storage
File Track

Customer Id	File Number	File Name	File Type	File Location	File Submitted Date	Description	
2	81234	Verification File	Account Verification	Branch 2	20-3-2016	Account Verification Details Submitted	Track Your File

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
background-color: #FFFFFF;
}
.style1 {
color: #FF0033;
font-size: 24px;
}
table tr td{font-size:16px;}
-->
</style></head>
```

```

<body>
<center>

<table width="1091" height="336" border="0">
  <tr>
    <td height="65" colspan="2"><% @ include file="header.jsp"%></td>
  </tr>
  <tr>
    <td width="168" align="left" valign="top"><p>&nbsp;</p>
    <% @ include file="user_menu.jsp"%> </td>
    <td width="900"><label>
      </label>

      <p align="center">
        <table width="750" height="89" border="3" align="center"
bordercolor="#000000" bgcolor="#9999CC">
          <tr>
            <td width="81"><div align="center"><strong>Customer
Id</strong></div></td>
            <td width="92"><div align="center"><strong>File
Number</strong></div></td>
            <td width="92"><div align="center"><strong>File
Name</strong></div></td>
            <td width="92"><div align="center"><strong>File
Type</strong></div></td>
            <td width="92"><div align="center"><strong>File
Location</strong></div></td>
            <td width="92"><div align="center"><strong>File Submitted
Date</strong></div></td>
            <td width="92"><div
align="center"><strong>Description</strong></div></td>
          <td></td>
        </tr>

<% @ include file="connect.jsp" %> <%
String id=(String)session.getAttribute("cliId");
ResultSet rs1=st.executeQuery("select * from file_detail where
customer_id='"+id+"'");
while(rs1.next())
{
  >
  <tr>
    <td><center><%=rs1.getString("customer_id")%> </center></td>
    <td><center><%=rs1.getString("id")%></center></td>
    <td><center><%=rs1.getString("file_name")%></center></td>

```

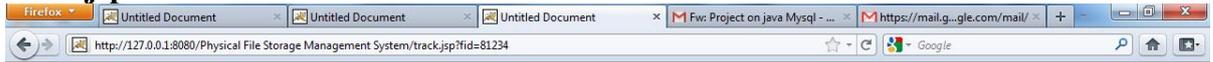
```
<td><center><%=rs1.getString("file_type")%> </center></td>
<td><center><%=rs1.getString("flocation")%> </center></td>
<td><center><%=rs1.getString("entryDate")%></center></td>
<td><center><%=rs1.getString("desc")%> </center></td>
<td><center><a href="track.jsp?fid=<%=rs1.getString("id")%>">Track Your
File</a> </center></td>
```

```
</tr>
<%
}
%>
</table>

</td>
</tr>
</table>
<div align="center"></div>

</center>
</body>
</html>
```

Projecthelpline.in



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

- [HOME](#)
- [ABOUT US](#)
- [CONTACT US](#)
- [FEEDBACK](#)
- [LOGOUT](#)
- [EMPLOYEE LOGIN](#)
- [ADMIN LOGIN](#)

Your Files History

- [Your Profile](#)
- [Files In Storage](#)
- [File Track](#)

Customer Id	File Number	File Name	File Type	File Location	File Submitted Date	Description	
2	81234	Verification File	Account Verification	Branch 2	20-3-2016	Account Verification Details Submitted	Not Assigned

Issued to Employee History

Employee Id	File Number	File Name	File Type	Status
E001	81234	Verification File	Account Verification	Completed

```

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
background-color: #FFFFFF;
}
.style1 {
color: #FF0033;
font-size: 24px;
}
.style3 {
font-size: 20px;
color: #FF0033;
}
table tr td{font-size:16px;}
    
```

```

-->
</style></head>

<body>
<center>

<table width="1091" height="336" border="0">
  <tr>
    <td height="65" colspan="2"><% @ include file="header.jsp"%></td>
  </tr>
  <tr>
    <td width="168" align="left" valign="top"><p>&nbsp;</p>
    <% @ include file="user_menu.jsp"%> </td>
    <td width="900"><label>
      </label>
      <p align="center" class="style3">Your Files History</p>
      <p align="center">
        <table width="750" border="2" align="center" bordercolor="#000000"
        bgcolor="#9999CC">
          <tr>
            <td width="81"><div align="center"><strong>Customer
            Id</strong></div></td>
            <td width="92"><div align="center"><strong>File
            Number</strong></div></td>
            <td width="92"><div align="center"><strong>File
            Name</strong></div></td>
            <td width="92"><div align="center"><strong>File
            Type</strong></div></td>
            <td width="92"><div align="center"><strong>File
            Location</strong></div></td>
            <td width="92"><div align="center"><strong>File Submitted
            Date</strong></div></td>
            <td width="92"><div
            align="center"><strong>Description</strong></div></td>
            <td></td>
          </tr>

          <% @ include file="connect.jsp" %> <%
          String cliId=(String)session.getAttribute("cliId");
          String fid=request.getParameter("fid");

          ResultSet rs1=st.executeQuery("select * from file_detail where
          customer_id='"+cliId+"' and id='"+fid+"'");
          while(rs1.next())
          {

```

```

String fstatus=rs1.getString("status"); %>
<tr>
<td><center><%=rs1.getString("customer_id")%> </center></td>
<td><center><%=rs1.getString("id")%></center></td>
<td><center><%=rs1.getString("file_name")%></center></td>
<td><center><%=rs1.getString("file_type")%> </center></td>
<td><center><%=rs1.getString("flocation")%> </center></td>
<td><center><%=rs1.getString("entryDate")%></center></td>
<td><center><%=rs1.getString("desc")%> </center></td>
<%
if(fstatus.equals("0"))
{
%>
<td style="color:#FFFF00"><center>Not Assigned </center></td>
<%
}
else
{
%>
<td style="color:#FF3333"><center>Assigned </center></td>

<%
}
%>

</tr>
<%
}
%>
</table>
<br />
<p align="center" class="style3">Issued to Employee History</p>
<p align="center">
<table width="750" border="2" align="center" bordercolor="#000000"
bgcolor="#9999CC">
<tr>
<td width="92"><div align="center"><strong>Employee
Id</strong></div></td>
<td width="92"><div align="center"><strong>File
Number</strong></div></td>
<td width="92"><div align="center"><strong>File
Name</strong></div></td>

```

```
<td width="92"><div align="center"><strong>File  
Type</strong></div></td>
```

```
<td width="92"><div align="center"><strong>Status</strong></div></td>
```

```
</tr>
```

```
<%
```

```
    ResultSet rsi=st.executeQuery("select * from issue where fid='"+fid+"' ");  
    while(rsi.next())  
    {
```

```
        String istatus=rsi.getString("status"); %>
```

```
        <tr>
```

```
        <tr>
```

```
        <td><center><%=rsi.getString("empid")%></center></td>
```

```
        <td><center><%=rsi.getString("fid")%></center></td>
```

```
        <td><center><%=rsi.getString("fname")%> </center></td>
```

```
        <td><center><%=rsi.getString("ftype")%> </center></td>
```

```
        <td><center><%=rsi.getString("status")%></center></td>
```

```
    </tr>
```

```
    <%
```

```
    }
```

```
    %>
```

```
</table>
```

```
</td>
```

```
</tr>
```

```
</table>
```

```
<div align="center"></div>
```

```
</center>
```

</body>

</html>

cli_profile.jsp



FILE STORAGE



PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

[HOME](#) [ABOUT US](#) [CONTACT US](#) [FEEDBACK](#) [LOGOUT](#) [EMPLOYEE LOGIN](#) [ADMIN LOGIN](#)

[Your Profile](#)

[Files In Storage](#)

[File Track](#)

Customer ID	Username	Password	Name	Company Name	Address	City	State	Email ID	Phone No.
2	Vikas1	123	Vikas Narang	infotech pvt. ltd.	Preet Vihar	Delhi	New Delhi	vicky@gmail.com	9876543212

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
    margin-left: 0px;
    margin-top: 0px;
    background-color: #FFFFFF;
}
.style1 {
    color: #FF0033;
    font-size: 24px;
}
table tr td{font-size:16px;}
-->
</style></head>
```

```

<body>
<center>

<table width="1091" height="336" border="0">
  <tr>
    <td height="65" colspan="2"><% @ include file="header.jsp"%></td>
  </tr>
  <tr>
    <td width="168" align="left" valign="top"><p>&nbsp;</p>
    <% @ include file="user_menu.jsp"%> </td>
    <td width="900"><label>
      </label>

      <p align="center">
        <table width="700" height="89" border="3" align="center"
bordercolor="#000000" bgcolor="#9999CC">
          <tr>
            <td width="81"><div align="center"><strong>Customer
ID</strong></div></td>
            <td width="92"><div
align="center"><strong>Username</strong></div></td>
            <td width="92"><div
align="center"><strong>Password</strong></div></td>
            <td width="92"><div align="center"><strong>Name</strong></div></td>
            <td width="92"><div align="center"><strong>Company
Name</strong></div></td>
            <td width="92"><div align="center"><strong>Address</strong></div></td>
            <td width="92"><div
align="center"><strong>City</strong></div></td>
            <td width="92"><div align="center"><strong>State</strong></div></td>

            <td width="100"><div align="center"><strong>Email
ID</strong></div></td>
            <td width="92"><div align="center"><strong>Phone
No.</strong></div></td>
          </tr>

          <% @ include file="connect.jsp" %> <%
String id=(String)session.getAttribute("cliId");
          ResultSet rs1=st.executeQuery("select * from customer where id='"+id+"'");
          while(rs1.next())
          {
            %>
            <tr>
              <td><center><%=rs1.getString("id")%> </center></td>
              <td><center><%=rs1.getString("username")%></center></td>

```

```
<td><center><%=rs1.getString("password")%> </center></td>  
<td><center><%=rs1.getString("name")%> </center></td>  
<td><center><%=rs1.getString("cname")%></center></td>  
<td><center><%=rs1.getString("address")%> </center></td>  
<td><center><%=rs1.getString("city")%> </center></td>  
<td><center><%=rs1.getString("state")%> </center></td>  
<td><center><%=rs1.getString("email")%> </center></td>  
<td><center><%=rs1.getString("phone")%></center></td>
```

```
</tr>  
<%=  
}<br>  
%>  
</table>
```

```
</td>  
</tr>  
</table>  
<div align="center"></div>  
</center>  
</body>  
</html>
```

Projecthelpline.in

show_feedback.jsp

FILE STORAGE

PHYSICAL FILE STORAGE MANAGEMENT SYSTEM

HOME ABOUT US CONTACT US FEEDBACK CUSTOMER LOGIN EMPLOYEE LOGIN LOGOUT

Admin DashBoard
Add Customer Details
Add Employee Details
Issue File to Employee

Admin Menu

FeedBack Details

ID	Name	Email ID	Phone No.	State	Comment
123	Naini Kaur	nain@gmail.com	0237509275	Delhi	Good File Management !!

```
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Untitled Document</title>
<style type="text/css">
<!--
body {
margin-left: 0px;
margin-top: 0px;
}
.style6 {
font-size: 24px;
color: #FF0033;
}
-->
</style></head>

<body>

<table width="1091" height="336" align="center">
<tr>
```

```

        <td height="65" colspan="2"> <p>&nbsp;</p>
        <% @ include file="header.jsp" %></td>
    </tr>
    <tr>
        <td width="168" valign="top"><p>&nbsp;</p>
        <% @ include file="menu.jsp" %></td>
        <td width="900">
            <p align="center"><a href="admin_menu.jsp">Admin Menu</a></p>
            <p align="center" class="style6">FeedBack Details </p>
            <table width="600" height="89" border="3" align="center" bordercolor="#000000"
            bgcolor="#9999CC">
                <tr>
                    <td width="81"><div align="center"><strong> ID</strong></div></td>
                    <td width="92"><div align="center"><strong>Name</strong></div></td>

                    <td width="100"><div align="center"><strong>Email
                    ID</strong></div></td>
                    <td width="92"><div align="center"><strong>Phone
                    No.</strong></div></td>
                    <td width="92"><div
                    align="center"><strong>State</strong></div></td>

                    <td width="92"><div
                    align="center"><strong>Comment</strong></div></td>

                </tr>

    <% @ include file="connect.jsp" %> <%
        ResultSet rs1=st.executeQuery("select * from feedback");
        while(rs1.next())
        {
            <%>
            <tr>
            <td><center><%=rs1.getString("id_no")%> </center></td>
            <td><center><%=rs1.getString("name")%></center></td>
            <td><center><%=rs1.getString("email_id")%> </center></td>
            <td><center><%=rs1.getString("phone_no")%> </center></td>
            <td><center><%=rs1.getString("state")%></center></td>
            <td><center><%=rs1.getString("comment")%></center></td>

            </tr>
            <%
                }
            <%>
        </table>
    </td>
    </tr>

</table>

```

```
</body>  
</html>
```

Projecthelpline.in

5.2 CODE EFFICIENCY

Reviewing of Code efficiency for a module is carried out after the module is successfully compiled and all the syntax errors eliminated. Code efficiency review is extremely cost-effective strategies for reduction in coding errors in order to produce high quality code. Normally, two types of efficiency are carried out on the code of a module - code optimization and code inspection. The procedure and final objective of these two efficiency techniques are very different as discussed below.

5.3 OPTIMIZATION OF CODE

Code optimization is an informal code analysis technique. In this technique, after a module has been coded, it is successfully compiled and all syntax errors are eliminated. Some members of the development team are given the code a few days before the optimization meeting to read and understand the code. Each member selects some test cases and simulates execution of the code by hand (i.e. trace execution through each statement and function execution). The main objectives of the optimization are to discover the algorithmic and logical errors in the code. The members note down their findings to discuss these in a optimization meeting where the coder of the module is also present.

Even though a code optimization is an informal analysis technique, several guidelines have evolved over the years for making this naïve technique more effective and useful. Of course, these guidelines are based on personal experience, common sense, and several subjective factors. Therefore, guidelines should be considered as examples rather than as rules to be applied dogmatically. Some of these guidelines are the following:

The team performing the code optimization should not be either too big or too small. Ideally, it should consist of three to seven members.

TESTING

Projecthelping.com

6.1 TESTING PHASES

Software Testing is an empirical investigation conducted to provide stakeholders with information about the quality of the product or service under test, with respect to the context in which it is intended to operate. This includes, but is not limited to, the process of executing a program or application with the intent of finding software bugs. It can also be stated as the process of validating and verifying that a software program/application/product meets the business and technical requirements that guided its design and development, so that it works as expected and can be implemented with the same characteristics.

A primary purpose for testing is to detect software failures so that defects may be uncovered and corrected. This is a non-trivial pursuit. Testing cannot establish that a product functions properly under all conditions but can only establish that it does not function properly under specific conditions.[11] The scope of software testing often includes examination of code as well as execution of that code in various environments and conditions as well as examining the aspects of code: does it do what it is supposed to do and do what it needs to do. In the current culture of software development, a testing organization may be separate from the development team. There are various roles for testing team members. Information derived from software testing may be used to correct the process by which software is developed.

Defects and failures

Not all software defects are caused by coding errors. One common source of expensive defects is caused by requirements gaps, e.g., unrecognized requirements, that result in errors of omission by the program designer. A common source of requirements gaps is non-functional requirements such as testability, scalability, maintainability, usability, performance, and security.

Software faults occur through the following process. A programmer makes an error (mistake), which results in a defect (fault, bug) in the software source code. If this defect is executed, in certain situations the system will produce wrong results, causing a failure.^[12] Not all defects will necessarily result in failures. For example, defects in dead code will never result in failures. A defect can turn into a failure when the environment is changed. Examples of these changes in environment include the software being run on a new hardware

platform, alterations in source data or interacting with different software.^[12] A single defect may result in a wide range of failure symptoms.

Compatibility

A frequent cause of software failure is compatibility with another application, a new operating system, or, increasingly, web browser version. In the case of lack of backward compatibility, this can occur (for example...) because the programmers have only considered coding their programs for, or testing the software upon, "the *latest* version of" this-or-that operating system. The unintended consequence of this fact is that: their latest work might not be fully compatible with earlier mixtures of software/hardware, or it might not be fully compatible with *another* important operating system. In any case, these differences, whatever they might be, may have resulted in (unintended...) software failures, as witnessed by some significant population of computer users.

This could be considered a "prevention oriented strategy" that fits well with the latest testing phase suggested by Dave Gelperin and William C. Hetzel, as cited below ^[13].

Input combinations and preconditions

A very fundamental problem with software testing is that testing under *all* combinations of inputs and preconditions (initial state) is not feasible, even with a simple product. This means that the number of defects in a software product can be very large and defects that occur infrequently are difficult to find in testing. More significantly, non-functional dimensions of quality (how it is supposed to *be* versus what it is supposed to *do*) -- for example, usability, scalability, performance, compatibility, reliability -- can be highly subjective; something that constitutes sufficient value to one person may be intolerable to another.

Static vs. dynamic testing

There are many approaches to software testing. Reviews, walkthroughs or inspections are considered as static testing, whereas actually executing programmed code with a given set of test cases is referred to as dynamic testing. The former can be, (and unfortunately in practice often is...) omitted, whereas the latter takes place when programs begin to be used for the first time - which is normally considered the beginning of the testing stage. This may actually begin before the program is 100% complete in order to test particular sections of

code (modules or discrete functions). For example, Spreadsheet programs are, by their very nature, tested to a large extent "on the fly" during the build process as the result of some calculation or text manipulation is shown interactively immediately after each formula is entered

1. UNIT TESTING:

This is the smallest testable unit of a computer system and is normally tested using the white box testing. The author of the programs usually carries out unit tests.

2. INTEGRATION TESTING:

In integration testing, the different units of the system are integrated together to form the complete system and this type of testing checks the system as whole to ensure that it is doing what is supposed to do. The testing of an integrated system can be carried out top-down, bottom-up, or big-bang. In this type of testing, some parts will be tested with white box testing and some with black box testing techniques. This type of testing plays very important role in increasing the systems productivity. We have checked our system by using the integration testing techniques.

3. SYSTEM TESTING:

A part from testing the system to validate the functionality of software against the requirements, it is also necessary to test the non-functional aspect of the system. Some examples of non-functional tools include tests to check performance, data security, usability/user friendliness, volume, load/stress that we have used in our project to test the various modules.

System testing consists of the following steps:

1. Program(s) testing.
2. String testing.
3. System testing.
4. System documentation.
5. User acceptance testing.

4. FIELD TESTING:

This is a special type of testing that may be very important in some projects. Here the system is tested in the actual operational surroundings. The interfaces with other systems and the real world are checked. This type of testing is very rarely used. So far our project is concerned; we haven't tested our project using the field testing.

5. ACCEPTANCE TESTING:

After the developer has completed all rounds of testing and he is satisfied with the system, then the user takes over and re-tests the system from his point of view to judge whether it is acceptable according to some previously identified criteria. This is almost always a tricky situation in the project because of the inherent conflict between the developer and the user. In this project, it is the job of the bookstores to check the system that whether the made system fulfills the goals or not.

WHY SYSTEM TESTING?

Testing is vital to the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved. Inadequate testing results in two types of problems:

1. The time lag between the cause and the appearance of the problem.
2. The effect of system errors on the files and records within the system.

6.2 ACTIVITY NETWORK FOR SYSTEM TESTING

The test plan entails the following activities:

1. Prepare test plan.
2. Specify conditions for user acceptance testing.
3. Prepare test data for program testing.
4. Prepare test data for transaction path testing.
5. Plan user training.
6. Compile/assemble programs.
7. Prepare job performance aids.
8. Prepare operational documents.

PREPARE TEST : A workable test plan must be prepared in accordance with established design specifications. It includes the following items:

- Outputs expected from the system.
- Criteria for evaluating outputs.
- A volume of test data.
- Procedure for using test data.
- Personnel and training requirements.

SPECIFY CONDITIONS FOR USER ACCEPTANCE TESTING

Planning for user acceptance testing calls for the analyst and the user to agree on conditions for the test.

PREPARE TEST DATA FOR PROGRAM TESTING

As each program is coded, test data are prepared and documented to ensure that all aspects of the program are properly tested.

PREPARE TEST DATA FOR TRANSACTION PATH TESTING

This activity develops the data required for testing every condition and transactions to be introduced into the system. The path of each transaction from origin to destination is carefully tested reliable results.

PLAN USER TRAINING

User training is designed to prepare the user for testing and converting the system. User involvement and training take place parallel with programming for three reasons:

- The system group has time available to spend on training while the programs are being written.
- Initiating a user-training program gives the systems group a clearer image of the user's interest in the new system.
- A trained user participates more effectively in system testing.

The training plan is followed by preparation of the user training manual and other text materials.

COMPILE / ASSEMBLE PROGRAMS

All programs have to be compiled / assembled for testing.

PREPARE JOB PERFORMANCE AIDS

In this activity the materials to be used by personnel to run the system are specified and scheduled. This includes a display of materials.

PREPARE OPERATIONAL DOCUMENTS

During the test plan stage, all operational documents are finalized including copies of the operational formats required by the candidate system.

SYSTEMS TESTING

The computer department to ensure that the system functions as specified does this testing. This testing is important to ensure that a working system is handed over to the user for acceptance testing.

ACCEPTANCE TESTING

The user to ensure that the system functions, as the user actually wanted performs this testing. With prototyping techniques, this stage becomes very much a formality to check the accuracy and completeness of processing. The screen layouts and output should already have been tested during the prototyping phase.

An error in the program code can remain undetected indefinitely. To prevent this from happening the code was tested at various levels. To successfully test a system, each condition, and combinations of conditions had to be tested. Each program was tested and linked to other programs. This unit of program is tested and linked to other units and so on until the complete system has been tested.

The purpose of testing is to ensure that each program is fully tested. To do so a test plan had to be created. The test plan consists of a number of test runs such as the valid paths through the code, and the exception and error handling paths. For each test run there is a list of conditions tested, the test data used and the result expected. The test plan was then reviewed to check that each path through the code is tested correctly. It is the responsibility of the programmer to collect the data that will produce the required test condition.

6.3 TEST CASES

		TEST DATA Specifications for Physical File Storage Management System user form1	
	Test Date	27-4-2010	Programmer name:
	Tested By:		Project ID:
			Physical File Storage Management System 11105454

Employee Name	<input type="text"/>	First Name	
Address	<input type="text"/>		the fields are required.Can enter only letters, spaces, hyphens, and apostrophes. No numeric & special characters are allowed(Length upto 32 characters) Will accept only letters, numbers, underscores, and one dot (.)
Age	<input type="text"/>		
BirthDay	<input type="text"/> Select Month	<input type="text"/> DD YYYY	Select here.
Country	<input type="text"/>		Select here
Recuirement	<input type="text"/>		
Password	<input type="text"/>		password length > than 6 alphanumeric
Re-Type Pasword	<input type="text"/>		password length > than 6 alphanumeric
Secret Question	<input type="text"/>		Select One
Your Answer	<input type="text"/>		Use 4 characters or more- not case sensitive

CREATE MY ACCOUNT

Positive Test cases for registration form

T.C ID	PRE-CONDITION	T.C DESCRIPTION	T.C DATA	EXPECTED	ACTUAL	RESULT
1	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Employee Name field	Akhila	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Ok	Pass
2	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Employee Name field	A S	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Ok	Pass
3	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Employee Name field	A Satish	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters. name are required.	Ok	Pass
4	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Name field	Akhila S	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Ok	Pass
5	User should be on https://Physical File Storage Management System /registration?And is on Employee Name Field	Check the functionality of Employee Name field	pooja Sharma	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Ok	Pass
6	User should be on https://Physical File Storage Management System /registration?And is on Address Field	Check the functionality of Address option	uttam nagar	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 132 characters. name are required.	Ok	Pass
7	User should be on https://Physical File Storage Management System /registration? And is on Address Field	Check the functionality of Address option	Female	Will accept one.This is required	Ok	Pass
8	User should be on https://Physical File Storage Management System /registration? And is on Address Field	Check the functionality of Address of sign up option	m-10/20 vikashpuri	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 132 characters. name are required.	Ok	Pass
9	User should be on https://Physical File Storage Management System /registration? And is on AgeField	Check the functionality of Age p option	101	Will accept only numeric upto 3 digit . Customer ID are required	Ok	Pass
10	User should be on https://Physical File Storage Management System /registration?And is on Birthday Field	Check the functionality of Birthday option	february 20 1995	Will accept all valid dates	Ok	Pass
11	User should be on https://Physical File Storage Management System /registration? And is on BirthdayField	Check the functionality of Birthday of sign up option	March 21 05	Will accept all valid dates	Ok	Pass

12	User should be on https://Physical File Storage Management System /registration? And is on Country DropDown List	Check the functionality of Country DropDown List	India	Will accept country.This is required.	Ok	Pass
13	User should be on https://Physical File Storage Management System /registration? And is on Postal Code Field	Check the functionality of Postal Code field	110075	Will accept the valid postal code of the selected country	Ok	Pass
14	User should be on https://Physical File Storage Management System /registration? And is on Requirement Field	Check the functionality of requirement of sign up option	Employee	Will accept only letters, numbers, underscores, and one dot (.)	Ok	pass
18	User should be on https://Physical File Storage Management System /registration? And is on Password Field	Check the functionality of password text box	Avhijklhm	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name ID.	Ok	Pass
19	User should be on https://Physical File Storage Management System /registration? And is on Password Field	Check the functionality of password text box	654321	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Ok	Pass
20	User should be on https://Physical File Storage Management System /registration? And is on Password Field	Check the functionality of password text box	2207**	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Ok	Pass
21	User should be on https://Physical File Storage Management System /registration? And is on Password Field	Check the functionality of password text box	Chinnu1312**	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Ok	Pass
22	User should be on https://Physical File Storage Management System /registration? And is on Retype Password field	Check the functionality of Retype password text box	Same as the password	Will accept the same password as above.	Ok	Pass
23	User should be on https://Physical File Storage Management System /registration? And is on Alternate Email Field	Check the functionality of Alternate Email text box	akhila.hunagund@gmail.com	Will accept any valid email id or blank	Ok	Pass
24	User should be on https://Physical File Storage Management System /registration? And is on Alternate Email Field	Check the functionality of Alternate Email test box	Blank	Will accept any valid email id or blank	Ok	Pass
27	User should be on https://Physical File Storage Management System /registration? And is on Secret Question field	Check the functionality of Secret Question	What was the make of your first car?	Either select one from drop list or type one	Ok	Pass
28	User should be on https://Physical File Storage Management System /registration? And is on Your Answer field	Check the functionality of Answer text box	Santro	Will accept letters and a single space only	Ok	Pass

Negative Test cases for registration form

T.C ID	PRE-CONDITION	T.C DESCRIPTION	T.C DATA	EXPECTED	ACTUAL	RESULT
1	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Name field	A	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Invalid	Fail
2	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Name field	Akhila's	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Invalid	Fail
3	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Name field	Satish	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Invalid	Fail
4	User should be on https://Physical File Storage Management System /registration? And is on Employee Name Field	Check the functionality of Name field	.	Will accept only letters, spaces, hyphens, and apostrophes.Length upto 32 characters.name are required.	Invalid	Fail
5	User should be on https://Physical File Storage Management System /registration? And is on Address Field	Check the functionality of Gender field	Space	Allows to select one.This field is required.	Invalid	Fail
6	User should be on https://Physical File Storage Management System /registration? And is on Age Field	Check the functionality of Birthday of sign up option	Numeric	Allows to select a month and enter a valid day and year	Invalid	Fail
7	User should be on https://Physical File Storage Management System /registration? And is on Birthday Field	Check the functionality of Birthday of sign up option	january 1 2011	Allows to select a month and enter a valid day and year	Invalid	Fail
8	User should be on https://Physical File Storage Management System /registration? And is on Birthday Field	Check the functionality of Birthday of sign up option	february 29 1995	Allows to select a month and enter a valid day and year	Invalid	Fail
9	User should be on https://Physical File Storage Management System /registration? And is on Country DropDown List	Check the functionality of Country DropDown List	-Select one-	Allows to select a country.This field is required.	Invalid	Fail
12	User should be on https://Physical File Storage Management System /registration? And is on Email Field	Check the functionality of ID of sign up option	Blank	Will accept only letters, numbers, underscores, and one dot (.).The ID must be available.	Invalid	Fail
13	User should be on https://Physical File Storage Management System /registration? And is on Email Field	Check the functionality of ID of sign up option	akhila@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.).The ID must be available.	Invalid	Fail

14	User should be on https://Physical File Storage Management System /registration? And is on Email Field	Check the functionality of ID of sign up option	#akhila@yahoo.com	Will accept only letters, numbers, underscores, and one dot (.) .The ID must be available.	Inva lid	Fail
15	User should be on https://Physical File Storage Management System /registration? And is on Password Field	Check the functionality of password text box	2#	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Inva lid	Fail
16	User should be on https://Physical File Storage Management System /registration? And is on Password Field	Check the functionality of password text box	a123	Will accept 6 to 32 characters.Capitalisation matters. and don't use your name or ID.	Inva lid	Fail
17	User should be on https://Physical File Storage Management System /registration? And is on Retype Password field	Check the functionality of Retype password text box	Different from the one in Password field	Will accept the same password as above	Inva lid	Fail
18	User should be on https://Physical File Storage Management System /registration? And is on Alternate Email Field	Check the functionality of Alternate Email text box	akhila.hunagund	Will accept any valid email id or blank	Inva lid	Fail
19	User should be on https://Physical File Storage Management System /registration? And is on Alternate Email Field	Check the functionality of Alternate Email test box	@yahoo.com	Will accept any valid email id or blank	Inva lid	Fail
22	User should be on https://Physical File Storage Management System /registration? And is on Secret Question 2 field	Check the functionality of Secret Question Drop Down List	-Select one-	Either select one from drop list or type one	Inva lid	Fail
23	User should be on https://Physical File Storage Management System /registration? And is on Your Answer field	Check the functionality of Answer text box	Blank	Will accept anything.But not blank	Inva lid	Fail

6.4 VERIFICATION AND VALIDATION (V&V)

The objectives of verification, validity activities are to assess and improve the quality of the work products generated during development and modification of the software. Quality depends upon the various attributes like correctness, completeness, consistency, reliability, usefulness, usability, efficiency and conformance to standards.

The terms verification and validation are used synonymously. These are defined as under: -

Verification: “Are we building the product right?”

Validation: “Are we building the right product?”

Verification activities include proving, testing, and reviews. Validation is the process of evaluating software at the end of the software development to ensure compliance with the software requirements. Testing is a common method of validation. Clearly, for high reliability we need to perform both activities. Together, they are often called V&V activities.

The major V&V activities for software development are inspection, reviews, and testing (both static and dynamic). The V&V plan identifies the different V&V tasks for the different phases and specifies how these tasks contribute to the project V&V goals. The methods to be used for performing these V&V activities, the responsibilities and milestones for each of these activities, inputs and outputs for each V&V task, and criteria for evaluating the outputs are also specified.

The two major V&V approaches are testing and inspections. Testing is an activity that can be generally performed only on code. It is an important activity and is discussed in detail in a later chapter. Inspection is a more general activity that can be applied to any work product, including code. Many of the V&V tasks are such that for them, an inspection type of activity is the only possible way to perform the tasks (e.g. trace ability and document evaluation). Due to this, inspections play a significant role in verification.

SYSTEM IMPLEMENTATION



Projectline.in

7. POST IMPLEMENTATION MAINTENANCE AND REVIEW

As we know, creating software is one thing and the implementation of the created software is another. The process of implementing software is much difficult as compared to the task of creating the project. First we have to implement the software on a small scale for removing the bugs and other errors in the project and after removing them we can implement the software on a large scale.

Before we think in terms of implementing the Software on a large basis, we must consider the Hardware requirements.

Whenever we develop software or project a certain hardware and software is being used by the programmer for developing the project. The hardware and software to be used by the programmer for developing the project should be such that it would result in the development of a project, which would satisfy all the basic needs for which the project has been created by the programmer. The Hardware should be such that cost constraints of the Client should also be taken into account without affecting the performance.

7.1 HARDWARE EVALUATION FACTORS

When we evaluate computer hardware, we should first investigate specific *physical and performance* characteristics for each hardware component to be acquired. These specific questions must be answered concerning many important factors. These *hardware evaluation factors* questions are summarized in the below figure.

Notice that there is much more to evaluating hardware than determining the fastest and cheapest computing device. For e.g. the question of possible obsolescence must be addressed by making a technology evaluation. The factor of *ergonomics* is also very important. Ergonomics is the science and technology that tries to ensure that computers and other technologies are "user-friendly", that is safe, comfortable and easy to use. *Connectivity* is another important evaluation factor, since so many computer systems are now interconnected within wide area or local area telecommunications networks.

HARDWARE EVALUATION FACTORS

- 1) Performance
- 2) Cost
- 3) Reliability
- 4) Availability
- 5) Compatibility
- 6) Modularity
- 7) Technology
- 8) Ergonomics
- 9) Connectivity
- 10) Environmental requirements
- 11) Software
- 12) Support

7.2 SOFTWARE EVALUATION FACTORS

Software can be evaluated according to many factors similar to the hardware evaluation. Thus the factors of *performance, cost, reliability, compatibility, modularity, technology, ergonomics, and support* should be used to evaluate proposed software acquisitions. In addition, however, *the software evaluation factors* are summarized in below figure. For e.g. some software packages require too much memory capacity and are notoriously slow, hard to use, or poorly documented. They are not a good selection for most end users, even if offered at attractive prices.

SOFTWARE EVALUATION FACTORS:

1. **EFFICIENCY:** is the software a well-written system of computer instructions that does not use much memory capacity or CPU time?
2. **FLEXIBILITY:** can it handle its processing assignments easily without major modifications?

3. **SECURITY:** does it provide control procedures for errors, malfunctions and improper use?
4. **LANGUAGE:** do our computer programmers and users write it in a programming language that is used?
5. **DOCUMENTATION:** is the s/w well documented? Does it include helpful user instructions?
6. **HARDWARE:** does existing hardware have the features required to best use this software?
7. Other characteristics of hardware such as its performance, what about the cost, how much is reliable and etc.

7.3 CONVERSION AND TRAINING

An important aspect of is to make sure that the new design is implemented to establish standards. The term implementation has different meanings, ranging from the conversion of a basic application to a complete replacement of a computer system. Implementation is used here to mean the process of converting a new or revise system into an operational one. Conversion is one aspect of implementation. Conversion means changing from one system to another. The objective is to put the tested system into operation while holding costs, risks, and personnel irritation to a minimum. It involves creating computer-compatible files, training the operation staff, and installing terminal and hardware. A critical aspect of conversion is not disrupting the functioning of the organization.

When a new system is used over and old, existing and running one, there are always compatibility errors. These errors are caused because of the lack of equipment or personnel to work the new system. Running any specified system at an organization does require some or other hardware or, in this case, software requirement as well.

Conversion is one aspect of implementation review & software maintenance.

There are three types of implementation:

1. Implementation of a computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verifying printouts for integrity.
2. Implementation of a new computer system to replace an existing one. This is usually a difficult conversion. If not properly planned there can be many problems. Some large computer systems have taken as long as year to convert.
3. Implementation of a modified application to replace an existing one, using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files.

7.4 TRAINING NEEDS

Training needs refer to the gaining of knowledge required for running the system.

First of all the system is a computer based system therefore the person should have good knowledge about computer and its working.

He should know how to use software's on the computer.

For a better usage and working of the software the organization should appoint a person who has good knowledge of all the required software. The organization gets a person trained through different institutes present in the market. The training should be as per the above requirements.

8. COST ESTIMATION OF THE PROJECT

Cost in a project is due to the requirements for software, hardware, and human resources. Hardware resources are computer time, terminal time and memory required for the project. Software resources include the tools and compilers needed during development. The bulk of cost of software development is due to human resources needed. Cost estimates are determined in terms of person-months (PM).

Total No. Of Persons Involved In This Project:

1. Administrator
2. Senior Programmer
3. Junior Programmers
4. On line Users.

Since this Project will complete in 4 months

COST ESTIMATE: (Salary of Project Manager + Salary of Senior Programmer + 2 * Salary of Junior Programmer) * 2

9. REPORT GENERATION

From above modules different reports are generated. Reports that can be generated :

- **Customer Reports:** Gives customers details.
- **Feedback Reports:** Gives customer feedback details.
- **File Details:** Gives file details, Issued file handled by the employee.
- **Employee Details:** Give employee details.
- **File Status:** Status of all the files that a company is handling.
- **File Issued Details:** Details of the issued files of customers.

Projecthelpline.in

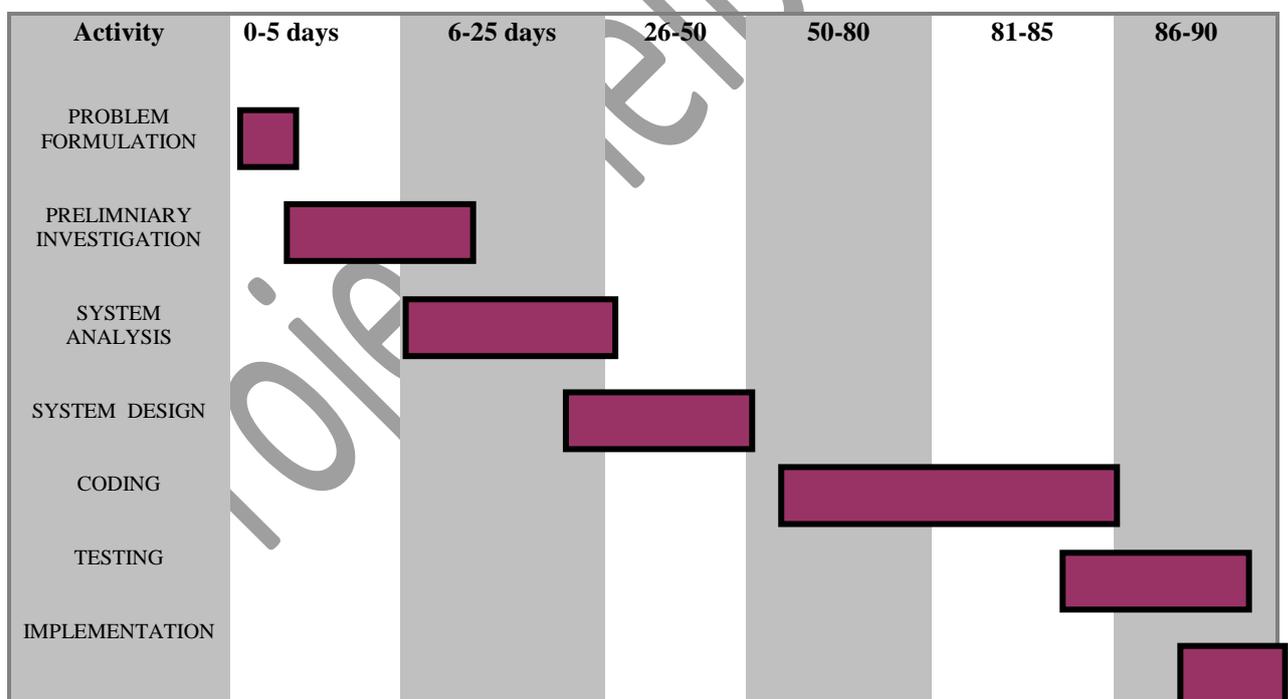
10. GANTT & PERT CHART

Gantt Chart

A **Gantt chart** is a graphical representation of the duration of tasks against the progression of time. A Gantt chart is a useful tool for planning and scheduling projects

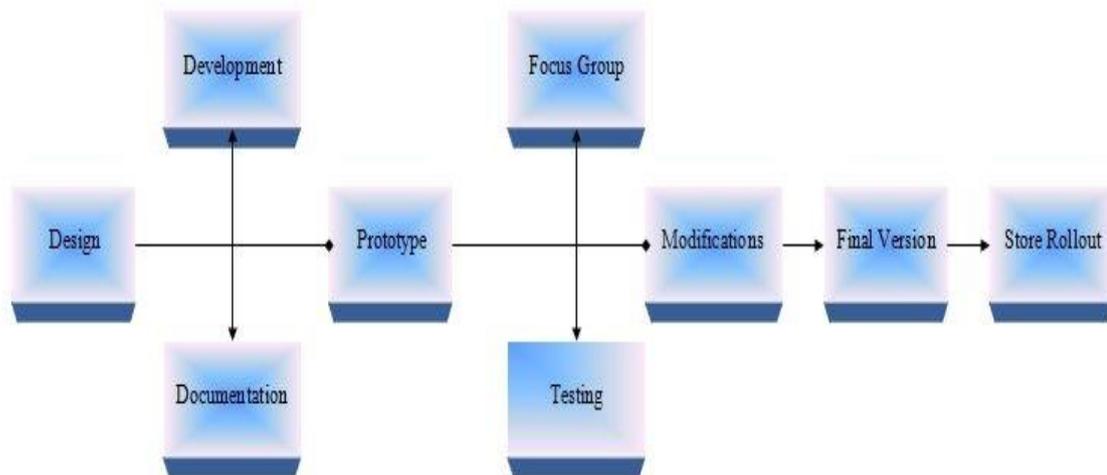
A Gantt chart is a type of bar chart that illustrates a project schedule. Gantt charts illustrate the start and finish dates of the terminal elements and summary elements of a project. Terminal elements and summary elements comprise the work breakdown structure of the project. Some Gantt charts also show the dependency.

Gantt chart is a project scheduling technique. Progress can be represented easily in a Gantt chart, by coloring each milestone when completed. The project will start in the month of December and end after 4 months at the end of March .



Pert Chart

PERT (Project Evaluation and Review Technique) charts consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies.



PERT is a method to analyze the involved tasks in completing a given project, especially the time needed to complete each task, and identifying the minimum time needed to complete the total project.

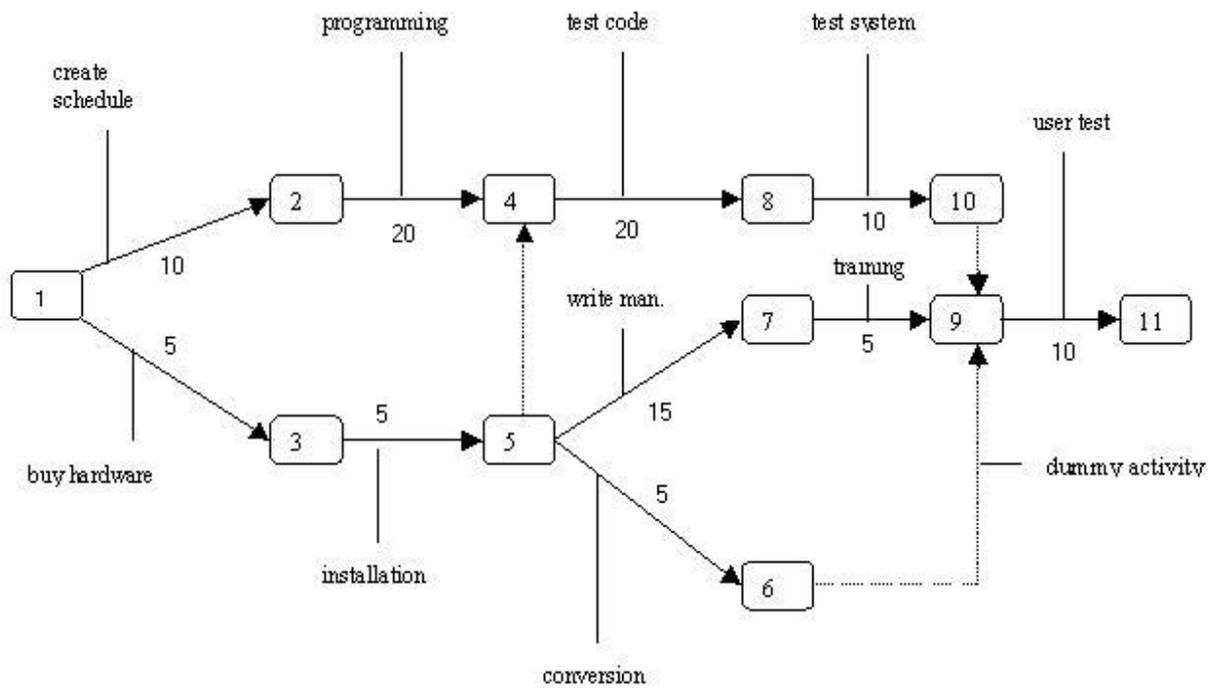
A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project. PERT stands for *Program Evaluation Review Technique*, a methodology developed by the U.S. Navy in the 1950s to manage the Polaris submarine missile program. A similar methodology, the *Critical Path Method* (CPM) was developed for project management in the private sector at about the same time. A critical path in a PERT chart is shown by using thicker arrows.

Steps in the PERT Planning Process

PERT planning involves the following steps:

1. Identify the specific activities and milestones.
2. Determine the proper sequence of the activities.

3. Construct a network diagram.
4. Estimate the time required for each activity.
5. Determine the *critical path*.
6. Update the PERT chart as the project progresses.



PERT Chart

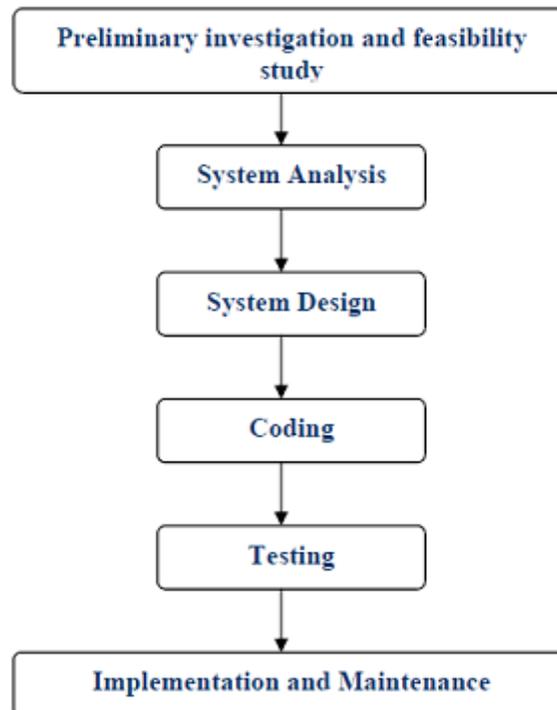
- * Numbered rectangles are nodes and represent events or milestones.
- * Directional arrows represent dependent tasks that must be completed sequentially.
- * Diverging arrow directions (e.g. 1-2 & 1-3) indicate possibly concurrent tasks
- * Dotted lines indicate dependent tasks that do not require resources.

FIGURE B: PERT Chart representation of the “PHYSICAL FILE STORAGE MANAGEMENT SYSTEM ”

PERT charts are a more sophisticated form of activity chart. In activity diagrams only the estimated task durations are represented. Since the actual durations might vary from the estimated durations, the utility of the activity diagrams is limited.

Project Planning and Scheduling

The PFSMS has gone through the following stages of development in its Software Development Life Cycle.



The following GANTT and PERT chart is showing the schedule of various stages of Physical File Storage Management System .

11. SECURITY AND VALIDATION CHECKS

In this project we have used following validation checks.

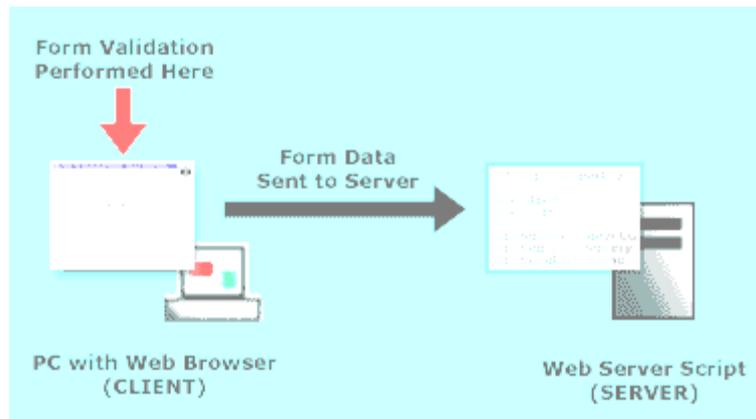
- While entering the data into the form it will check for the name of the client is properly filled & it should not be null.
- Whenever we enter the data for the new customer, company, or user will automatically check the details from the database tables and also generate the connection number automatically.
- Similarly in the complaint table complaint number will generate automatically.
- Entered text / number should not exceed the limit (width).
- Almost for all fields we have used the validation for example if name of the fields requires the text type of data then it will check for the string and if the data is numeric then it will check if the number entered is proper numeric or not.

‘Verification & Validation Model’ makes it sure that, certain rules are followed at the time of development of a software product and also makes it sure that the product that is developed fulfills the required specifications. This reduces the risk associated with any software project up to certain level by helping in detection and correction of errors and mistakes, which are unknowingly done during the development process.

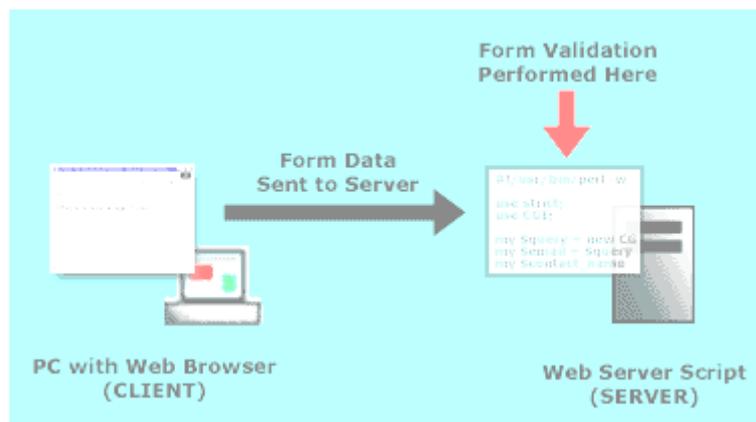
What is form validation?

Form validation is the process of checking that a form has been filled in correctly before it is processed. For example, if your form has a box for the user to type their email address, you might want your form handler to check that they've filled in their address before you deal with the rest of the form.

There are two main methods for validating forms: *server-side* (using CGI scripts, ASP, etc), and *client-side* (usually done using JavaScript). Server-side validation is more secure but often more tricky to code, whereas client-side (JavaScript) validation is easier to do and quicker too (the browser doesn't have to connect to the server to validate the form, so the user finds out instantly if they've missed out that required field!).



Client-side form validation (usually with JavaScript embedded in the Web page)



Server-side form validation (usually performed by a CGI or ASP script)

Project

12. SCOPE OF FUTURE APPLICATION



This project is designed to meet the requirements of the users for software testing. It has been developed in JSP, keeping in mind the specifications of the system.

For designing the system we have used simple data flow diagrams.

Overall the project teaches us the essential skills like:

- ❑ Using system analysis and design techniques like data flow diagram in designing the system.
- ❑ Understanding the database handling and query processing using My Sql.

13. BIBLIOGRAPHY

- [1] Herbert Scheldt, **Java Complete Reference**, Fifth Edition, Tata McGraw Hill Edition.
- [2] Phil Hanna, **JSP 2.0: The Complete Reference**, Tata McGraw Hill Edition, 2003.
- [3] Elmarsi and Navathe, **Fundamentals of Database System** (Third Edition), Addison Wesley.
- [4] Ian Somerville, **Software Engineering**, Third Edition, Pearson Education.
- [5] Ali Bahrami, **Object-Oriented System Development**, Third Edition, Tata McGraw Hill Edition.
- [6] Ivan Bayross, **SQL, PL/SQL programming language of Oracle**, Second Edition, BPB Publication.

WEB REFERENCES

- [1] www.google.com
- [2] www.htmlcodetutorial.com

Projecthelpline.in

14. APPENDICES

TEXT BOX

A text Box control, sometimes called an edit field or edit control, displays information entered at design time, entered by the user, or assigned to the control in code at run time.

BUTTON

Use a Command Button control to begin, interrupt, or end a process. When chosen, a Command Button appears pushed in and so is sometimes called a push button.

LIST BOX

A List Box control displays a list of items from which the user can select one or more. If the number of items exceeds the number that can be displayed, a scroll bar is automatically added to the List Box control.

LABEL

A Label control is a graphical control you can use to display text that a user can't change directly.

GROUPBOX

A GROUPBOX control provides an identifiable grouping for controls. You can also use a Frame to subdivide a form functionally – for example, to separate groups of Option Button controls.

COMBO BOX

A Combo Box control combines the features of a text box and a list box. This control allows the user to select an item either by typing text into the combo box, or by selecting it from the list.

TIMER

A Timer control can execute code at regular intervals by causing a Timer event to occur. The Timer control, invisible to the user, is useful for background processing.

PICTURE BOX

The primary use for the Picture Box control is to display a picture to the user. The actual picture that is displayed is determined by the picture property. The picture property contains the file name (and optional path) for the picture file that you wish to display.

DATA GRID CONTROL

The DATA Grid control displays and operates on tabular data. It allows complete flexibility to sort, merge, and format tables containing strings and pictures. When bound to a Data control, MSFlexGrid displays read-only data.

DATE AND TIME PICKER CONTROL

A Date and Time Picker (DTP) Control provides a simple and intuitive interface through which to exchange data and time information with a user. For example, with a DTP control you can ask the user to enter a data and then retrieve his or her selection with ease.

OPTION BUTTON

An Option Button control displays an option that can be turned on or off.

IMAGE CONTROL

Use the Image control to display a graphic. An Image control can display a graphic from an icon, bitmap or metafile, as well as enhanced metafile, JPEG, or GIF files.

CHECK BOX CONTROL

A Check Box indicates whether a particular condition is on or off. We use check boxes in an application to give users true/false or yes/no options. Because check boxes work independently of each other, a user can select any number of check boxes at the same time.