Chapter 8

Web enables application and specification of the developed
Bibliographical Database System

8.1 Preamble.

Web based computing, the most important new structure in computing technology is
discussed in this chapter. The web enable aspect of Acquisition and Web Public Access
Catalogue (WPAC) of BDMS model is also studied in this chapter e.g. book selection
process, book request management, ordering, creation of vendor database,
accessioning, receiving and accounting. While writing the code for book order relation to
vender, already described in chapter seven, htp.p as a function is proposed which in
turn displays the captured data as a markup structure display.

This structure is now being applied to database applications and represents a shift way
from the traditional client/server mainframe architecture. The Web applications have
changed the rules of library use by increasing end-user access to mission-critical
applications, such as web renewal of book from borrowers desktop. A typical database
application on the web consists of four components.

- A web browser.
- A HTTP server.
- Database connection software.
- A database server.

The application interacts with the database, using the web as a mean of connection.
The web browser acts as the front end. It used HTML forms for queries and syntax
rules that define the formatting of documents on the web\(^1,3,5,10,13\). Web database
applications eliminate the installation of dedicated software on client machines which is
presently mandatory in case of almost all available LMS in India. For the designed
BDMS the only piece of software required to run web based applications is a web browser. Traditional database technology often requires users to learn a new interface for each new application. The Web database applications make a common graphical user interface (GUI) a reality. User need to learn only one interface, that of the web browser.

Applications Architecture:

Web enable database need a set of basic building blocks or layer.

- Browser layer.
- Applications logic layer.
- Database connection layer.
- Database layer.

The web browser is generally employed as the front end of web applications. The web browser is the client of a web enable database application. The application logic layer consists where the majority of an application’s functionality will be located. The application logic is responsible for

- Obtaining data for a query.
- Preparing the query and sending it to the database.
- Retrieving the results and formatting then for display.

The function of a database connection tool is to provide access to the underlying database which is also fast flexible reliable and easy to use.
The database is responsible for
❖ Storing data.
❖ Retrieving data based on user queries.
❖ Maintaining security for the housed data.

Essentially, all web enable applications do pretty much the same things.

8.1.1 Query Interface.

Web Applications provide users with an interface for entering data. The data they enter is usually called a "query" or a "request" because the user-defined data is used to dynamically query or make a request for some service on the web server machine (searching a database, ordering a book, requesting a file). (Appendix-I)

Transmit User-Defined Query - Once collected, the user-data is sent to a web server
Perform Server Side Processing - The web server processes the user-data using some sort of "middleware" e.g. PL./SQL or Declarative Language.

Massage Data - Processing almost always involves playing with data on the server. The user-defined request specifies how the data should be played with, 25,30,33,34.

8.1.2 Transmit Query Results.

The processed data is now returned to the client. Perform Client Side Processing - Finally, the returned data is displayed to the user. Display might be as simple as interpreting HTML, or as complex as performing calculations, sorting, or other manipulations of the data.
In the situation of web enable application work flow diagram.

1. User defines a Query using the Query interface
2. Query Interface sends the query to a server-side processing agent
3. Server-side agent responds to the query using some data source or other backend service
4. DataSource returns query results
5. Server-side agent returns query results
6. Query results are displayed to the user

Figure 8.1- Work flow diagram in web enabled BDMS environment.

For a straight HTML request in which a user
- instructs a web browser to
- contact a web server using the HTTP protocol, and
- ask it for a specific
- HTML document which the server
- returns to be
- displayed by the web browser.

For more complex example let a user supplied search parameters via
- HTML form that the web browser
- sent to a web server using an SSL connection. The web server might
- process that data with a CGI script by connecting to a
- data file, searching it based upon the user query, and
- sending back the results to be
- Displayed in the web browser window as a table.

This pattern could be made much more complex of course. For example, let a
1) Java client which
2) sends an encrypted request to a Java Web Server
3) which processes the request with a Java Servlet that uses a CORBA object to systemically generate
4) financial trade data in the form of XML with an associated XSL style sheet that is
5) sent back to the Java client to be
6) parsed into an XML Tree and displayed according to the XSL.

It is observed, in the web application workflow, motley of technologies can be used to perform various tasks. In fact, there are literally hundreds of technologies and tools used throughout this simple process. The sheer number of pieces is why many developers feel overwhelmed when they look at the processes of web application development. However, regardless of how many technologies are involved in a project, for the most part they can all be placed into one of four conceptual categories. That is, a technology will fall into the category of 1) the display layer (GUI), 2) the communication layer, 3) the middleware layer, or 4) the data layer.

![Diagram of four conceptual categories of web application]

Figure 8.2- Four conceptual categories of web application.
In the chapter five for acquisition model, approval request will be accepted from the faculty member is proposed. So the faculty member should have the access to fill up the requisition form from his desktop computer. As because BDM could not be access without a user_id and a password therefore a general user is created, with only privilege to insert in the reqbk table. During the study a relation reqbk is proposed with TEMP table space and any user can access the relation with URL http://kamal/WebDb/WebDb_Book_req_form.show, it seeks User_id and the passwd if supplied then the following form will be display on the user’s desktop.

![Figure 8.4- Book Requisition Form.](image)
8.2 NODAL grammar proposed during my study.

Procedure

<table>
<thead>
<tr>
<th>Description</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>To print a strong bold text</td>
<td>htp.bold('Hello...!')</td>
</tr>
<tr>
<td>To send a standard string of text to the output</td>
<td>htp.print(string)</td>
</tr>
<tr>
<td>Format the enclose string with emphasis</td>
<td>htp.emphasis(string)</td>
</tr>
<tr>
<td>Formats the output string in monospace formatting (showing code or instruction)</td>
<td>htp.keyboard</td>
</tr>
<tr>
<td>To create &lt;strong&gt; tag</td>
<td>htp.strong</td>
</tr>
<tr>
<td>Insert new line</td>
<td>htp.br</td>
</tr>
<tr>
<td>Hard return</td>
<td>htp.hr</td>
</tr>
<tr>
<td>New line</td>
<td>htp.line</td>
</tr>
<tr>
<td>For a &lt;HTML&gt; tag</td>
<td>htp.htmlOpen</td>
</tr>
<tr>
<td>For a &lt;/HTML&gt; tag</td>
<td>htp.htmlClose</td>
</tr>
<tr>
<td>Insert a &lt;HEAD&gt; tag</td>
<td>htp.headOpen</td>
</tr>
<tr>
<td>Insert a &lt;/HEAD&gt; tag</td>
<td>htp.headClose</td>
</tr>
<tr>
<td>Insert a &lt;BODY&gt; tag</td>
<td>htp.bodyOpen</td>
</tr>
<tr>
<td>Insert a &lt;/BODY&gt; tag</td>
<td>htp.bodyClose</td>
</tr>
<tr>
<td>Insert a &lt;TITLE&gt; tag</td>
<td>htp.title(string)</td>
</tr>
</tbody>
</table>

htp.meta(descriptionstring, name, content);

htp.meta(Refresh, Null,'5');- (automatically refresh the page after every five minutes)

htp.meta(Refresh, Null,'5';"http://kamal/WebDB/WebDB.home");

htp.fontOpen (color,fontface,size,attributes);

htp.img(pathOfImage, alignment, alternate,Ismap,attributes);

htp.mailto(emailaddress,text, name, attributes);

Source code of the procedure is provided in appendix-II

144
8.3 Reference.

1. AKEROYD, JOHN and COX, ANDREW. Integrated Library Management Systems: overview. VINE; Vol.29, 2; 1999

2. ANSARI, MEHTAB and ALAM. AMITA Awareness and use of OPACs in five Delhi libraries. In The Electronic Library; Vol.26, 1; 2008

3. ARFIELD, JOHN., BROWN, JEFF., BURTON, JIM., and WALLIS, RICHARD. Developing a World-Wide Web OPAC. VINE; Vol.25, 2; 1995.


6. BROWN, CHRISTOPHER. MEAGHER, and ELIZABETH S. Cataloging free e-resources: is it worth the investment? Inter lending & Document Supply; vol. 36,3; 2008.


10. COX, ANDREW. Library services on the Web. VINE; Vol.28, 4; 1998.

13. EBENEZER, CATHERINE. Trends in integrated library systems. VINE; Vol.32, 4; 2002
20. HENRY, MARCIA. Library OPAC vendors and the Web: Innopac, Ameritech, DRA
   Campus-Wide information Systems; vol.13, 2; 1996 Research paper.
21. HENRY, MARCIA. OPACS, Libraries, Universities: WWW directories. Campus-
   Wide Information Systems; Vol.12, 4; 1995.


26. KHURSHID, ZAHIRUDDIN AHMED, SYED SAJJAD. From online catalogs to library portals: empowering users. VINE; vol.37 3; 2007.

27. LOMBARDO, KRISTINE, SHAWN V., CONDIC. S. Empowering users with a new online catalog. Library Hi Tech; Vol.18, 2; 2000.


29. MARCAS, JAIR DE., BRANSE, YOSEF., GOLAN, YAEL. IGRA, IRIS. Hybrid library development at the University of Haifa Library. Library Review; vol. 49, 4; 2000.


31. RHYNO, ARTHUR. From library systems to mainstream software: how Web technologies are changing the role of the systems librarian. In Library Hi Tech; Vol.21, 3; 2003.

