

INTRODUCTION

This is a plan to make tender publishing online. As a part of it, tender data is recorded along with particulars of the products, their specifications and conditions. Tenders published will be visible to the visitors of the site till the closing date of tender.

Many huge enterprises handle large/huge projects of government and other private enterprises. In all such projects thousands of employees would be working on them. Such enterprise requires purchases from various vendors for different purposes. The employees would float tenders globally by advertising in reputed news papers. Advertising in news papers involves lot of cost, searching in news papers is time-consuming, communication gap between buyers and suppliers (only way is through phone or mail), receiving bids and finalizing tenders takes a lot of time. So we make this whole process of tendering online which is efficient and consumes very less time .

PROBLEM IDENTIFICATION

Existing System:-

Present System: Many huge enterprises handle large/huge projects of government and other private enterprises. In all such projects thousands employees would be working on them. Such enterprise requires purchases from various vendors for different purposes. The employees would float tenders globally by advertising in reputed news papers.

Disadvantages:

- Advertising in news papers involves lot of cost.
- Once the paper is lost, you cannot view them.
- Searching in news papers is time-consuming.
- Communication gap between buyers and suppliers(only way is through phone or mail)

- Receiving bids and finalizing tenders takes a lot of time.

Proposed System:-

Here, the entire process of tendering is done online.

- Buyers publish tenders.
- Tenders consist of product name, its specifications, quantity etc.
- Suppliers log in and view the tender details.
- If the supplier is new, he registers in the website.
- Supplier can pose queries if he has any.
- Buyers answer the queries posed by suppliers.
- If interested, supplier bids for the product. Buyers view the bid and select the best bid.

Advantages:

- No cost for advertising in news papers.
- Tenders can be viewed anywhere and anytime.
- Suppliers can ask queries immediately.
- Bidding through online saves a lot of time.
- Buyers can answer queries posed by suppliers immediately.

FEASIBILITY STUDY

The next step in analysis is to verify the feasibility of the proposed system. “All projects are feasible given unlimited resources and infinite time”. But in reality both resources and time are scarce. Project should be confirmed to time bound and should be optimal in their consumption of resources. This places a constant approval of any project.

Feasibility which is applied to **E-Tendering** pertains to the following areas:

- Technical feasibility
- Operational feasibility
- Economical feasibility

TECHNICAL FEASIBILITY:

To determine whether the proposed system is technically feasible, we should take into consideration the technical issues involved behind the system. **E-Tendering** uses the web technologies, which is rampantly employed these days worldwide. The world without the web is incomprehensible today. That goes to proposed system technically.

OPERATIONAL FEASIBILITY:

To determine the operational feasibility of the system we should take into consideration the awareness level of the users. This system is operational feasible since the users are familiar with the technologies and hence there is no need to gear up the personnel to use system. Also the system is very friendly and to use.

ECONOMIC FEASIBILITY:

To decide whether a project is economically feasible or not, we have to consider various factors as:

- Cost benefit analysis
- Long-term returns
- Maintenance cost

- The proposed **E-Tendering** is computer based. It requires average computing capabilities and access to internet, which are very basic requirements and can be afforded by any organization hence it doesn't incur additional economic overheads, which renders the system economically feasible.

SYSTEM ANALYSIS

SOFTWARE REQUIREMENT SPECIFICATION:-

- What is SRS?

Software Requirement Specification (SRS) is the starting point of the software developing activity. As system grew more complex it became evident that the goal of the entire system cannot be easily comprehended. Hence the need for the requirement phase arose. The software project is initiated by the client needs.

The SRS is the means of translating the ideas of the minds of clients (the input) into a formal document (the output of the requirement phase.)

The SRS phase consists of two basic activities:

- 1) **Problem/Requirement Analysis:** The process is order and more nebulous of the two, deals with understand the problem, the goal and constraints.
- 2) **Requirement Specification:** Here the focus is on specifying what has been found giving analysis such as representation, specification languages and tools, and checking the specifications are addressed during this activity.

Document Conventions:

We have used Times New Roman (text size 12). Bold Font is used for Main Headings (text size of 20). Bold, Underline and Italicized font is used for Modules (text size of 14). Underlining is done for technically important words.....

Intended Audience and Reading Suggestions:

This document is for better understanding of E-TENDERING for various Mailing technologies. Mainly intended for Hod sir, Internal guide, Staff members, Users and colleagues. This detail given below guides every normal user to how to go through this document for better understanding .The sequence to follow for better understanding is here Purpose, Scope , Features of E-TENDERING Software, Operating requirements, Modules present in the project, Advantages, References etc...

The Requirement phase terminates with the production of the validate SRS document. Producing the SRS document is the basic goal of this phase.

ROLE OF SRS:

The purpose of the Software Requirement Specification is to reduce the communication gap between the clients and the developers. Software Requirement Specification is the medium through which the client and user needs are accurately specified. It forms the basis of software development. A good SRS should satisfy all the parties involved in the system.

SCOPE:

This software is used for tender management and posted tenders can be viewed anywhere and anytime. Suppliers can ask queries immediately. Bidding online saves a lot of time and buyers can answer queries posted by suppliers immediately.

DATA FLOW DIAGRAMS

INTRODUCTION:

The system specification is produced at the culmination of the analysis task. The function and Performance allocated the software as part of the system engineering are refined by the established a complete information of performance requirements and design constraints, appropriate validation criteria, and others data pertaining to requirements. The tools that are used in this system analysis are data flow diagram and data dictionary.

DATA FLOW DIAGRAMS:

A Data Flow Diagram (DFD) is a graphical technique that depicts information flow and the transforms that are applied as data move from input to output.

Data flow diagram is a logical model of a system. The model does not depend on hardware, software, and data structure or file organization. It only shows the data flow between modules to module of the entire system. Data flow diagrams can be completed using only four notations as follows,

Data Flow Diagrams are of two types:

- 1) **Physical Data Flow Diagrams:** These are implementation-dependent i.e., they show the actual devices, departments, people, etc., involved in the system.
- 2) **Logical Data Flow Diagrams:** These diagrams describe the system independently of how it is actually implemented, they show what takes places, rather than how an activity is accomplished.

- **Data Flow:** Data move in a specific direction from an origin to destination. The data flow is a “packet” of data.

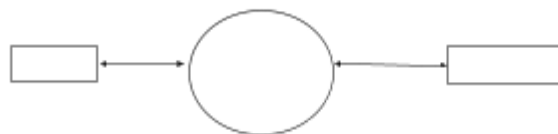


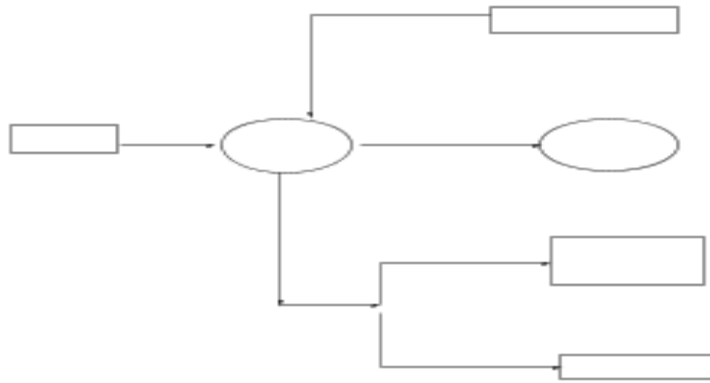
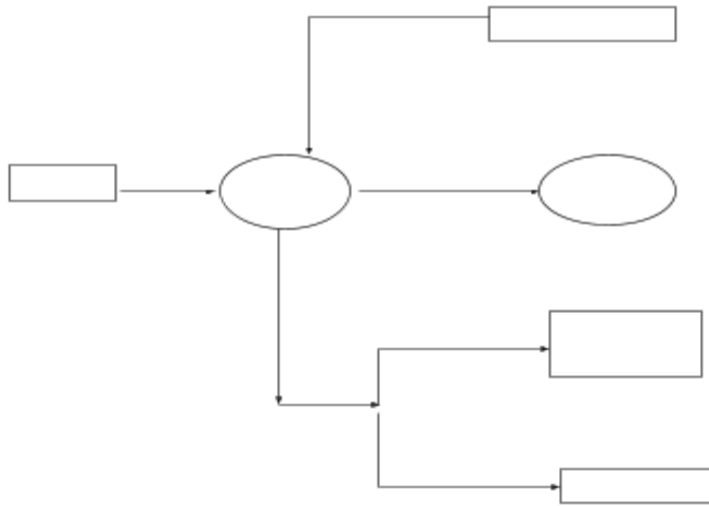
- **Process:** People, procedures or devices that produce data. The physical component is not identified.

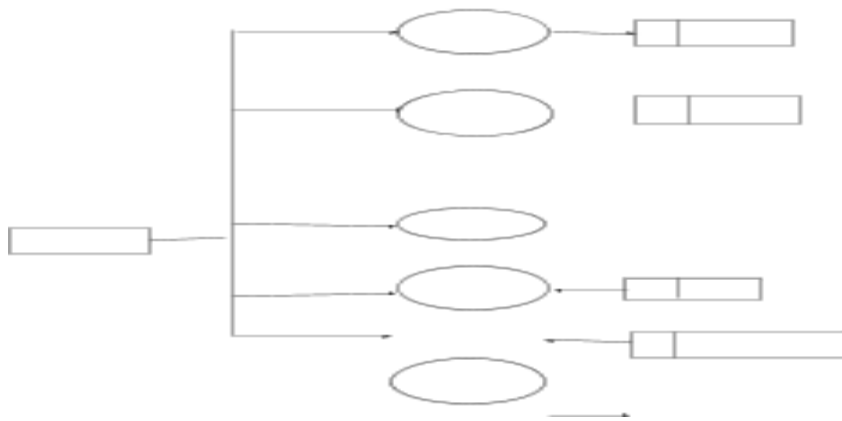


- **Source or Destination of Data:** External sources or destinations of data, which may be people or organizations or other entities.

- **Data Source:** Here a process  data in the system

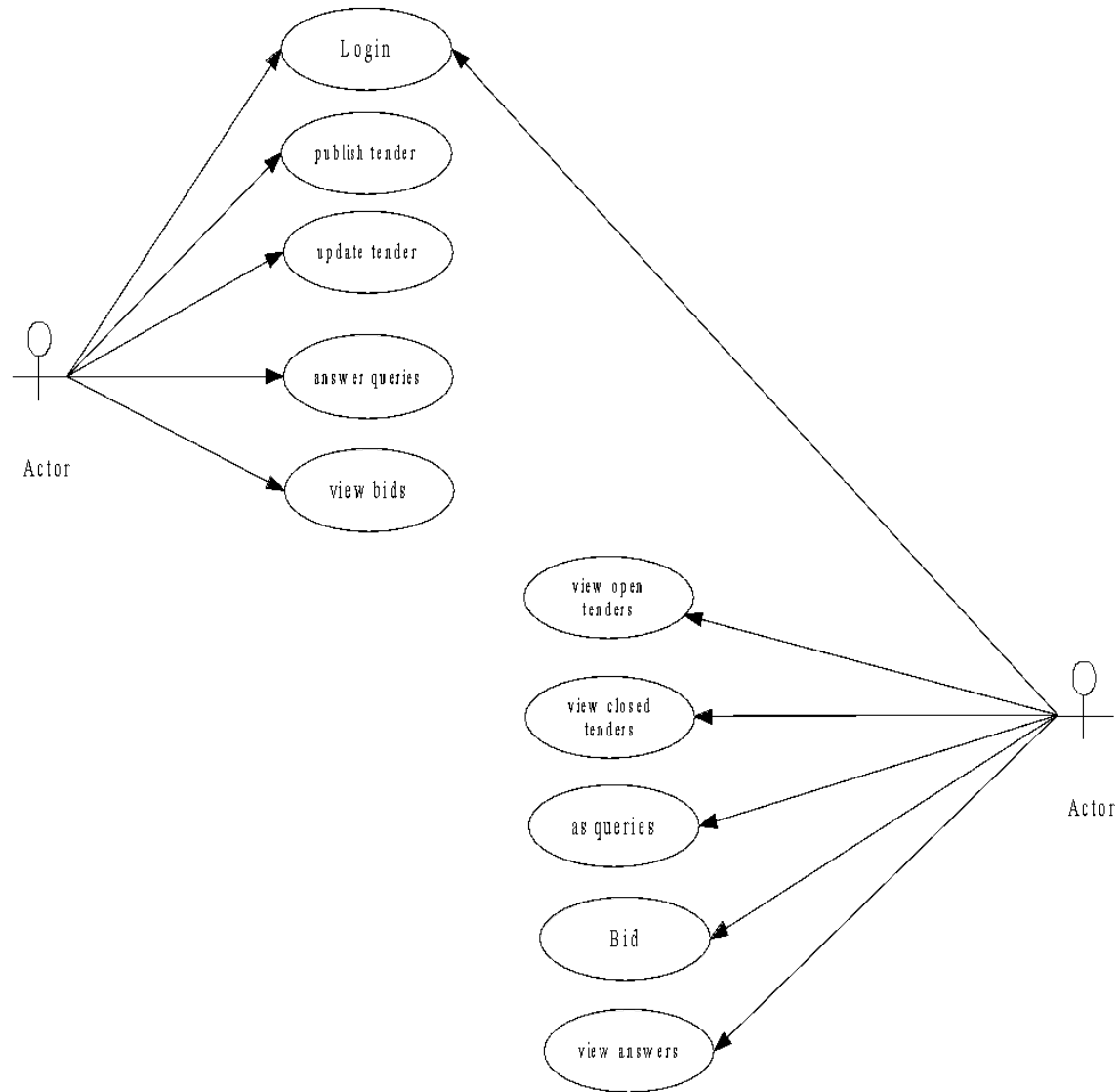






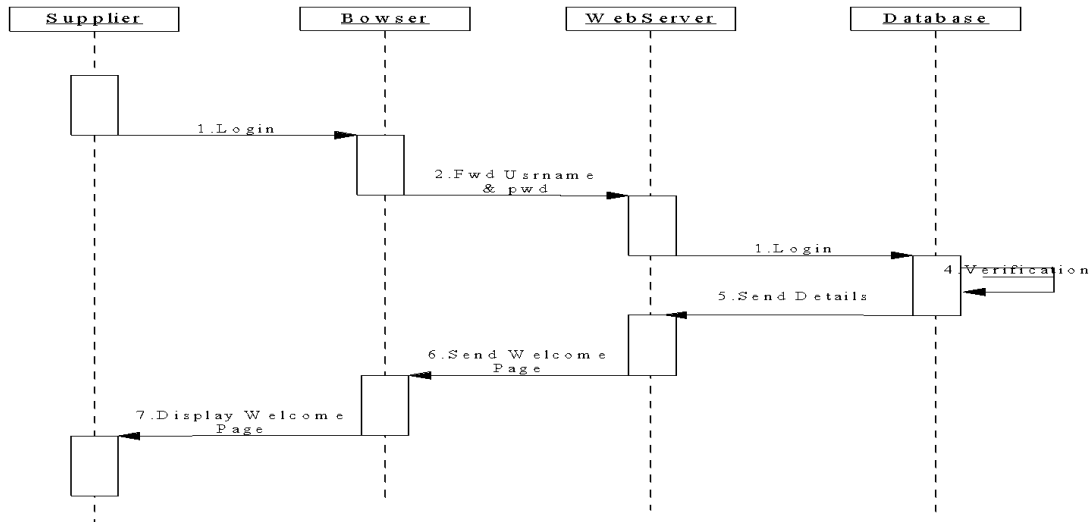
UML

Use-Case:

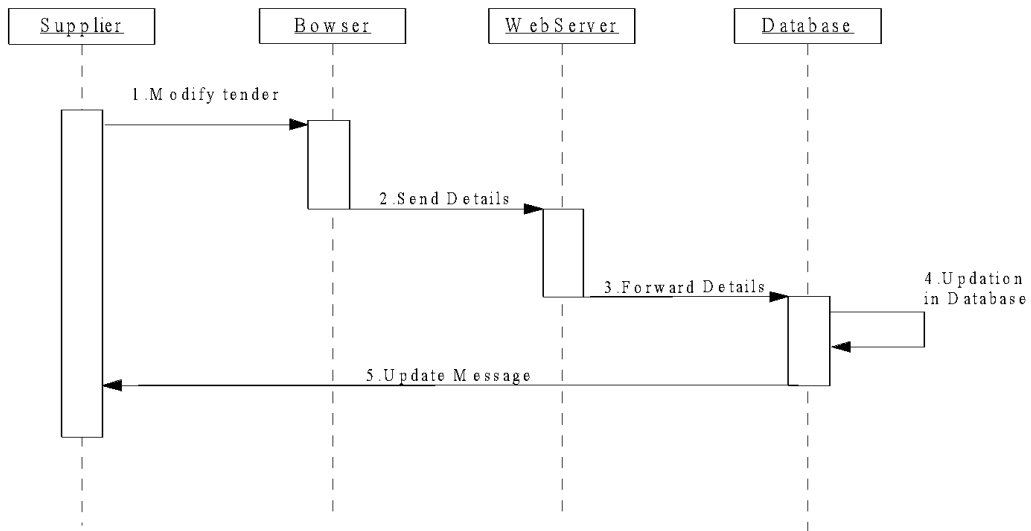


Sequence Diagrams:-

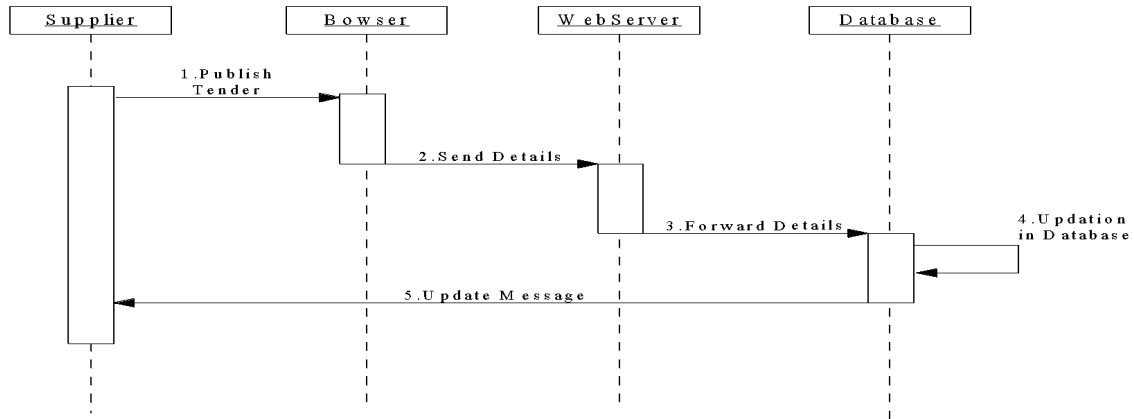
Supplier Login:



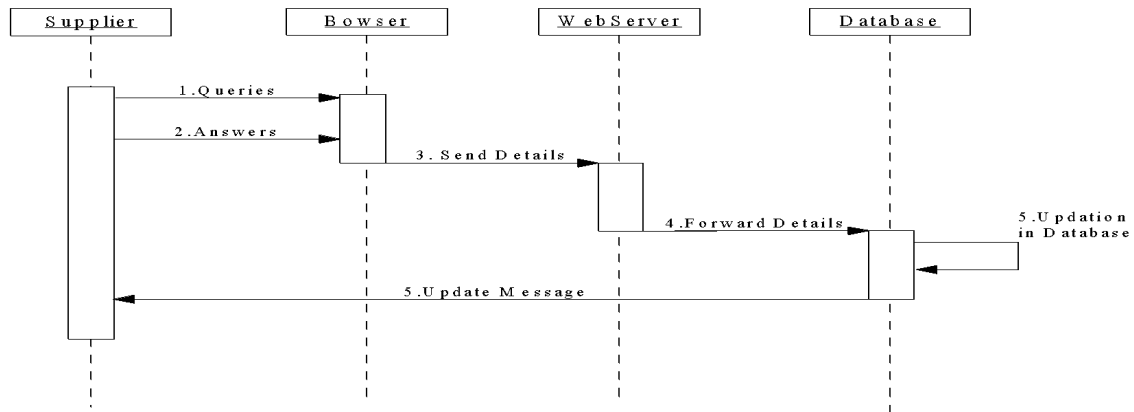
Admin Modify Tender:



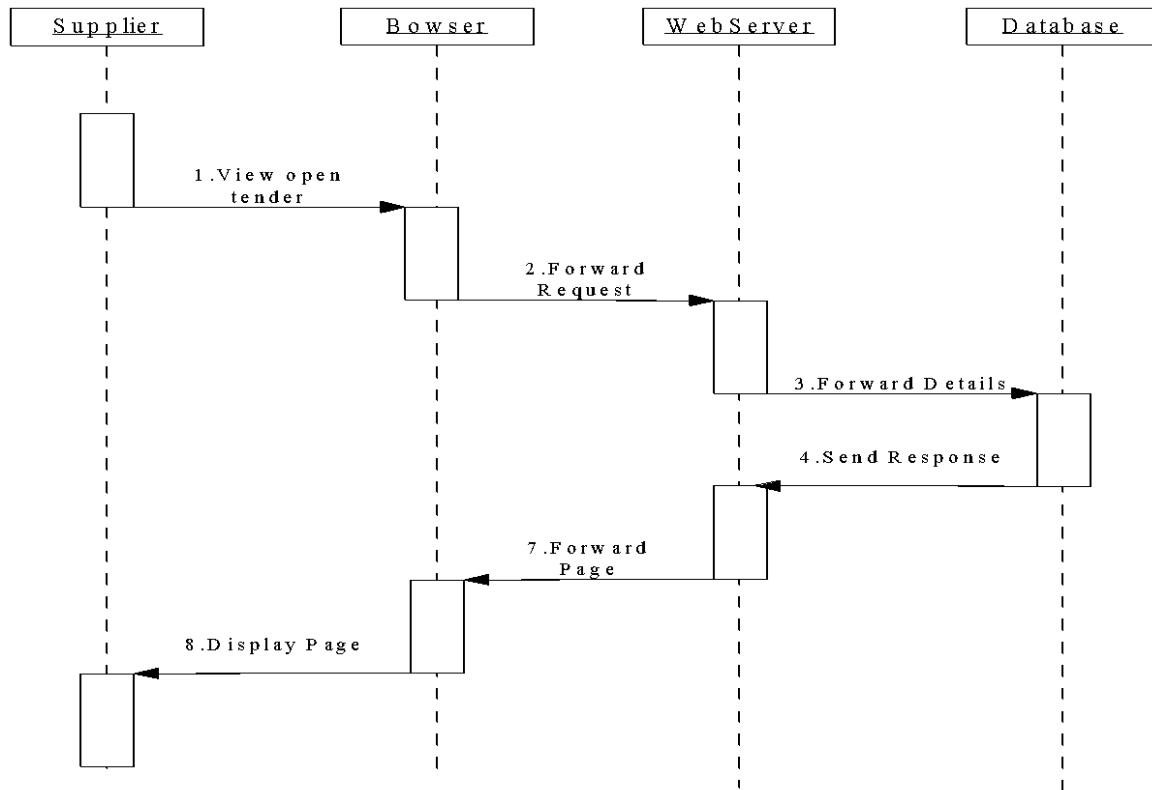
Admin publish tender:



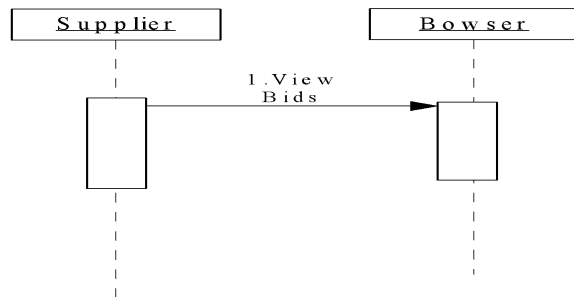
Admin view Queries & Answers:



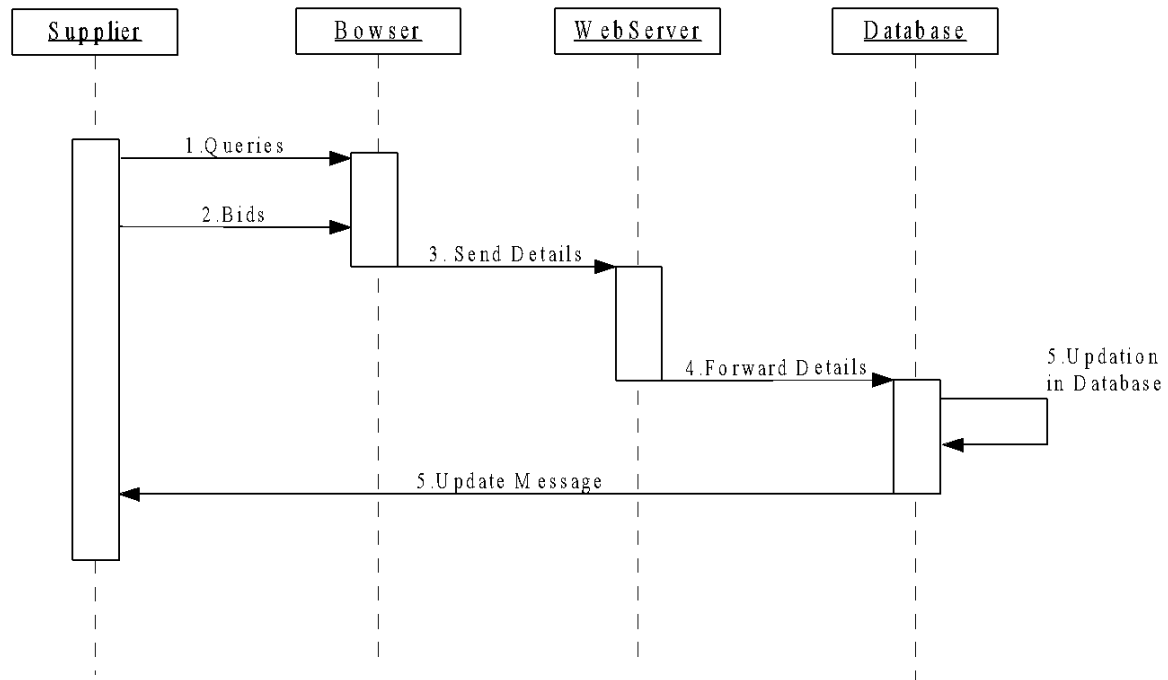
Supplier view open tenders:



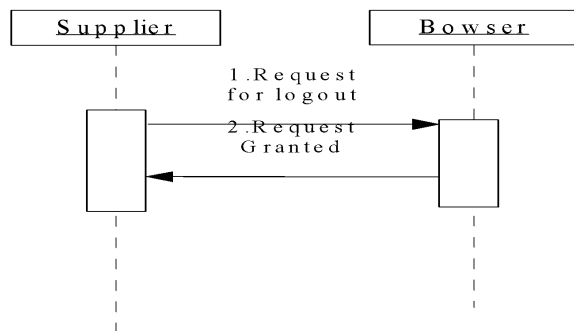
View Bid:



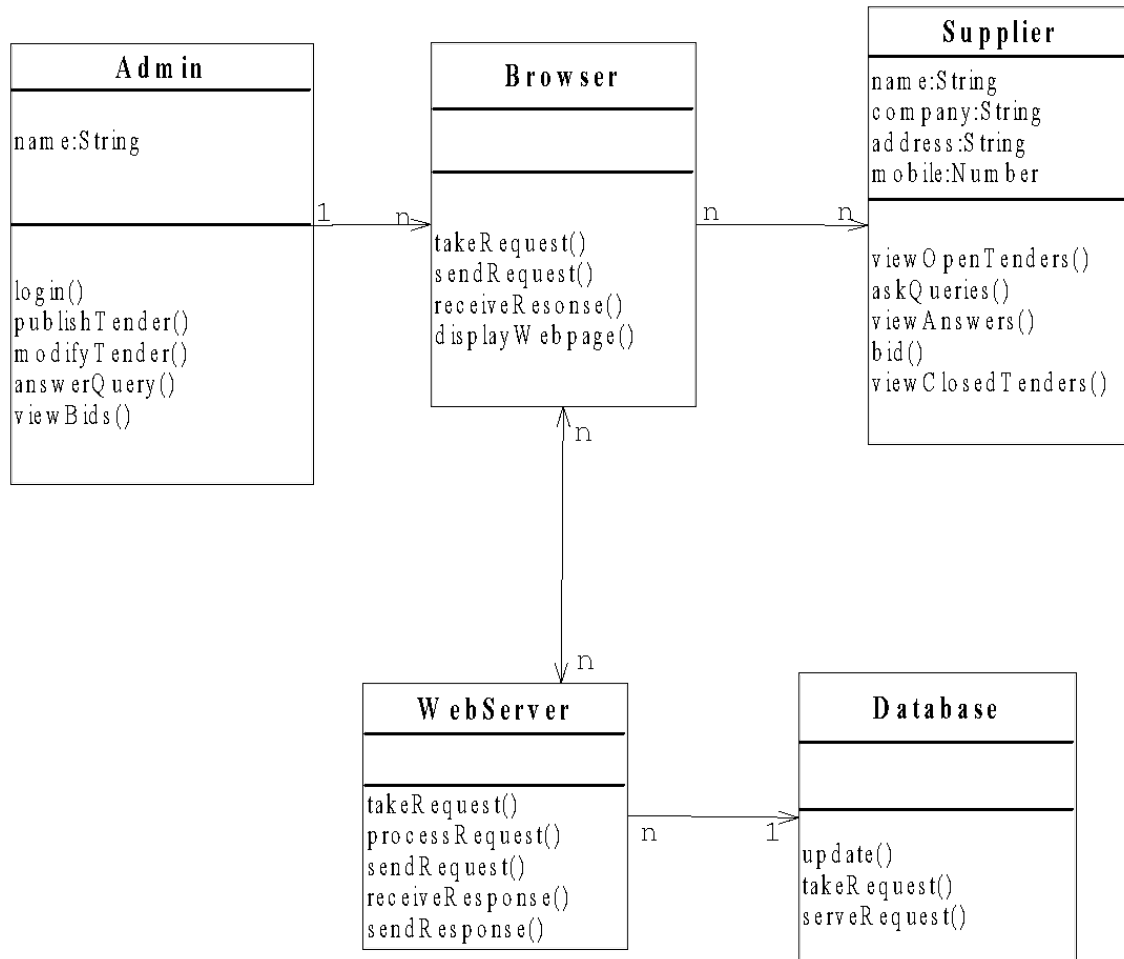
View Query & Bid:



Logout:



Class Diagram:



HARDWARE AND SOFTWARE REQUIREMENTS

Hard Ware Requirements:-

- **Processor:: Pentium-II(or) Higher**

- Ram:: 512MB (or) Higher
- Cache:: 512MB
- Hard disk:: 10GB

Soft Ware Requirements:-

- Web Server : Tomcat 5.5
- Server-side Technologies : Java, Java Server Pages
- Client-side Technologies : Hyper Text Markup Language, Cascading Style Sheets, Java Script, AJAX
- Database Server : MS Access
- Operating System : Windows (or) Linux (or) Mac any version

SYSTEM DESIGN

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare

input and output specification, details of implementation plan and prepare a logical design walkthrough.

The database tables are designed by analyzing functions involved in the system and format of the fields is also designed. The fields in the database tables should define their role in the system. The unnecessary fields should be avoided because it affects the storage areas of the system. Then in the input and output screen design, the design should be made user friendly. The menu should be precise and compact.

SOFTWARE DESIGN :-

In designing the software following principles are followed:

1. **Modularity and partitioning:** Software is designed such that, each system should consists of hierarchy of modules and serve to partition into separate function.
2. **Coupling:** Modules should have little dependence on other modules of a system.
3. **Cohesion:** Modules should carry out in a single processing function.
4. **Shared use:** Avoid duplication by allowing a single module be called by other that need the function it provides.

MODULE DESIGN:-

The major modules of the project are:

- Security and Accounts module
- Product Management module
- Tender Management module
- Querying and Bidding Module

Security and Accounts Module:

This specification is provided with privacy to prevent access to the unauthorized user

accounts. This can be enhanced with a Pretty Good Privacy to provide security to various accounts.

Product Management module:

Here the Administrator publishes tenders online. As a part of it, tender data is recorded along with the particulars of products, their specifications and conditions. Tenders recorded will be visible to the visitors of the site on a particular date as mentioned in the data.

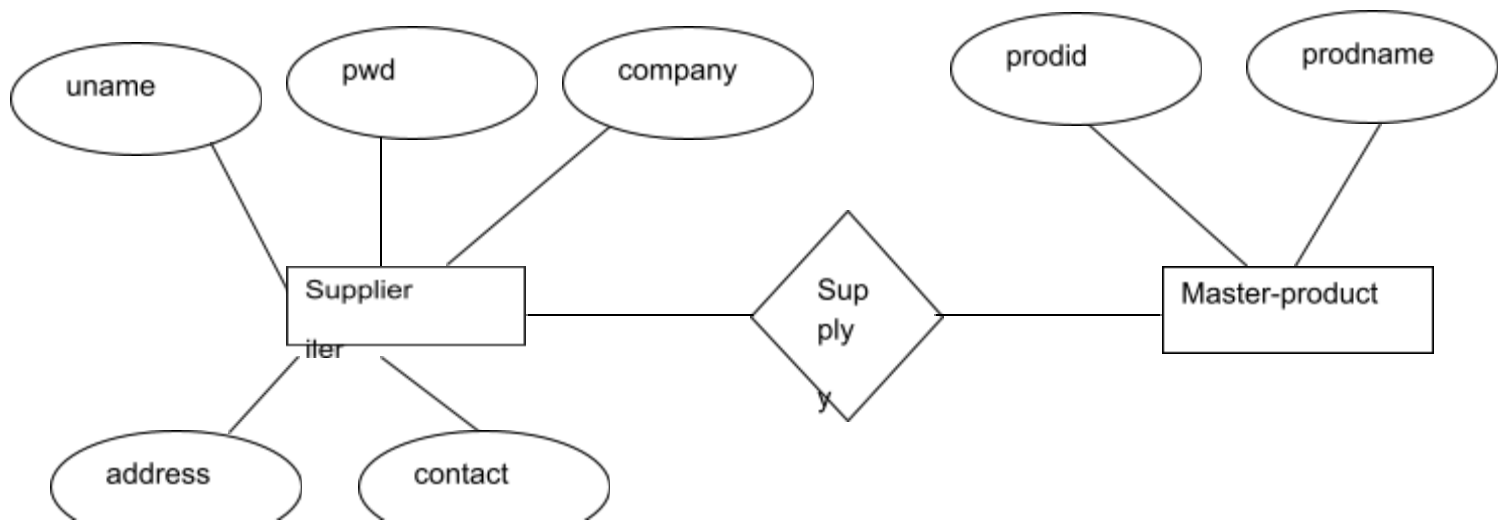
Tender Management module:

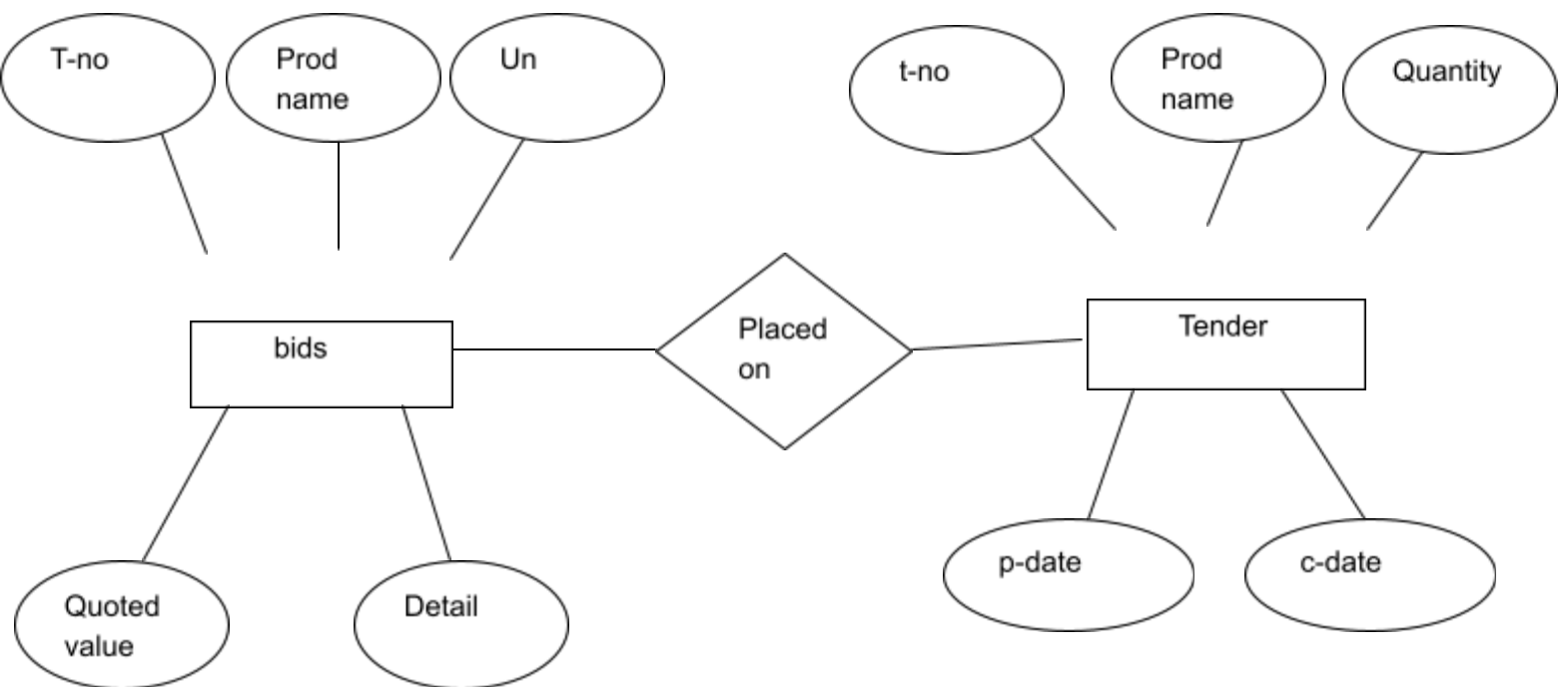
Here the administrator modifies the tender if required, that is the particulars of products their specifications and conditions can be modified.

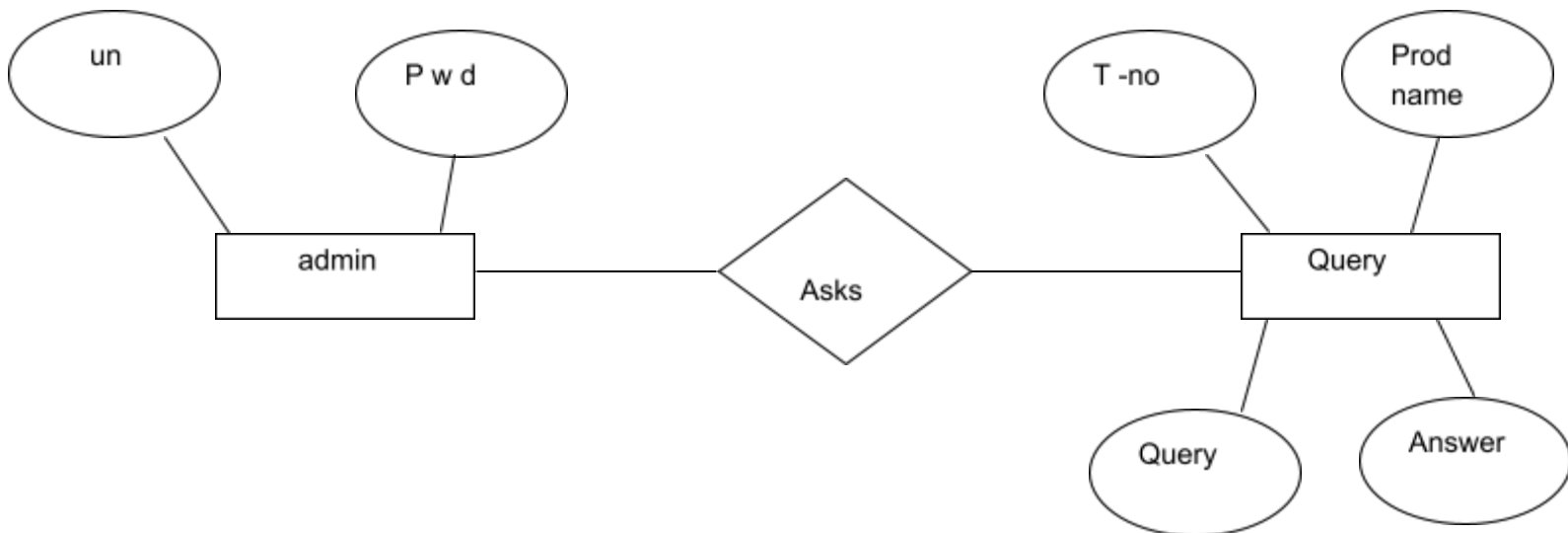
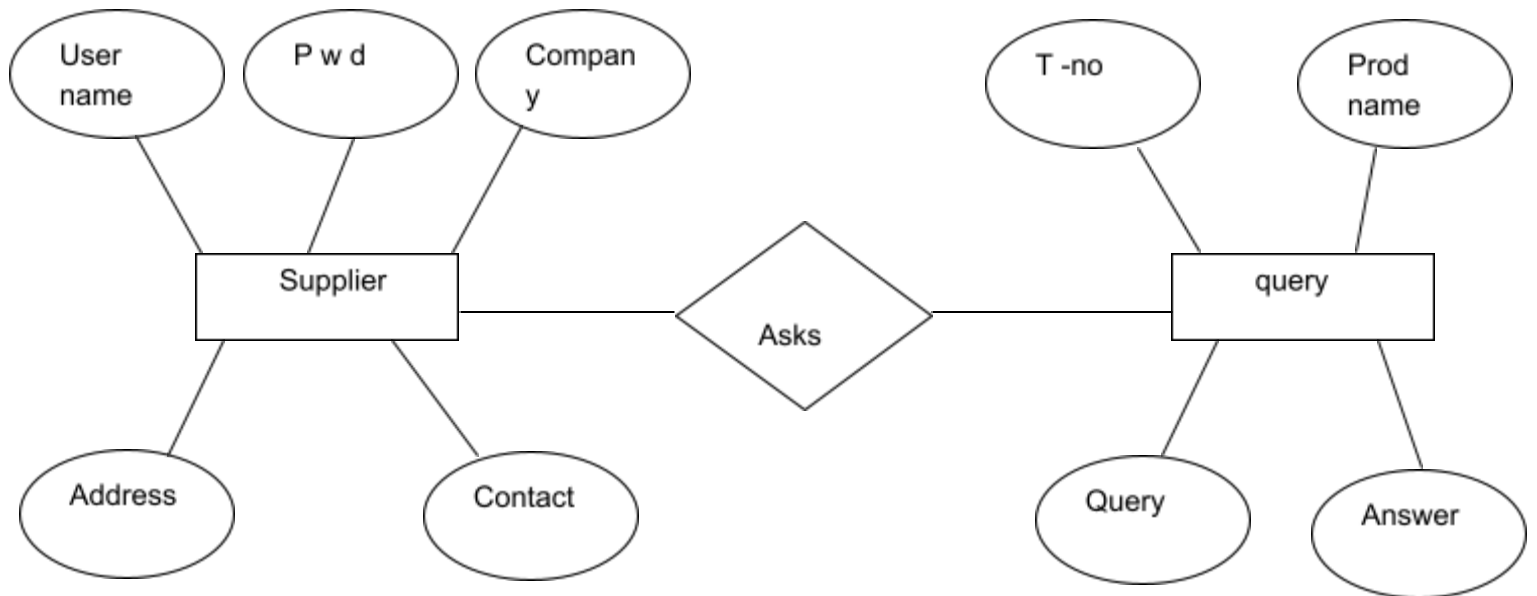
Querying and Bidding Module:

Here suppliers login and view the tenders details. If the suppliers are new, he/she registers in the website. Suppliers can pose queries if he/she has any. Then buyers answer the queries posed by the suppliers. Here if interested, suppliers bid the product. Buyers view the bid and select the best bid

DATABASE DESIGN







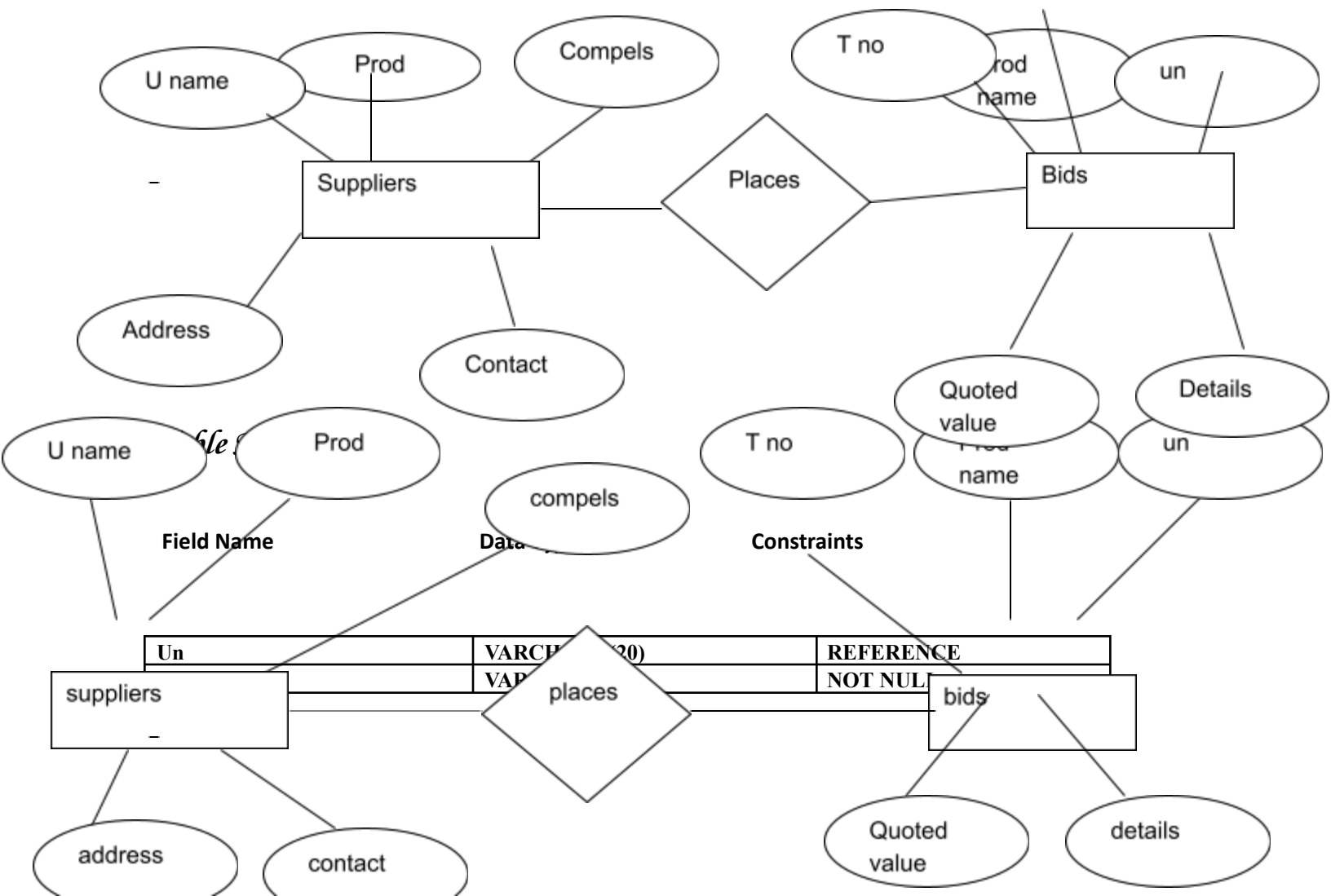
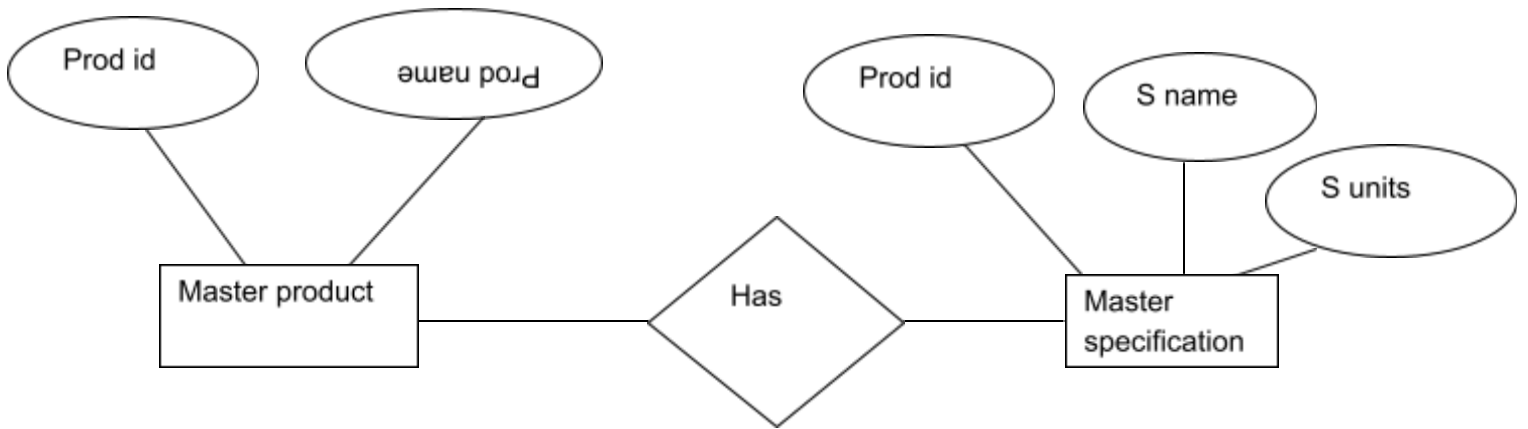


Table Name: suppliers

FieldName

Data Type

Constraints

Un	NUMBER(4)	NOT NULL
pwd	VARCHAR2 (20)	NOT NULL
company	VARCHAR2 (20)	NOT NULL
address	VARCHAR2 (20)	PRIMARY KEY
contact	VARCHAR2 (20)	NOT NULL

Table Name: master_product

Field Name

Data Type

Constraints

Product id	VARCHAR2 (20)	REFERENCE
Product_name	VARCHAR2 (1)	NOT NULL

Table Name : master_specification

Field Name

Data Type

Constraints

Product_id	VARCHAR2 (100)	NOT NULL
S_name	DATE	NOT NULL
S_units	DATE	NOT NULL

Table Name : tender

Field Name Data Type Constraints

T no	DATE	NOT NULL
Prod name	DATE	NOT NULL
quantity	VARCHAR2 (10)	NOT NULL
P date	VARCHAR2 (20)	REFERENCE
C date	VARCHAR2(20)	REFERENCE

Table Name : tender_specification

Field Name Data Type Constraints

T no	VARCHAR2 (20)	REFERENCE
Prod name	VARCHAR2 (10)	NOT NULL
S name	VARCHAR2 (20)	REFERENCE
S value	VARCHAR2 (500)	NOT NULL
units	VARCHAR2 (10)	NOT NULL

Table Name: query

Field Name Data Type Constraints

T no	VARCHAR2 (20)	REFERENCE
Prod name	VARCHAR2 (500)	NOT NULL
query	VARCHAR2 (2000)	NOT NULL
Answer	VARCHAR2 (20)	REFERENCE

Table Name : bids

Field Name Data Type Constraints

T_no	NUMBER(3)	COMPOSITE PRIMARY KEY
Prod_name	VARCHAR2 (20)	
un	VARCHAR2 (500)	NOT NULL
Quoted_value	VARCHAR2 (50)	NOT NULL
detail	VARCHAR2 (50)	NOT NULL

INPUT/OUTPUT DESIGN

Input design: considering the requirements, procedures to collect the necessary input data in most efficiently designed. The input design has been done keeping in view that, the interaction of the user with the system being the most effective and simplified way. Also the measures are taken for the following

- Controlling the amount of input
- Avoid unauthorized access to the classroom.
- Eliminating extra steps
- Keeping the process simple
- At this stage the input forms and screens are designed.

Output design: All the screens of the system are designed with a view to provide the user with easy operations in simpler and efficient way, minimum key strokes possible. Instructions and important information is emphasized on the screen. Almost every screen is provided with no error and important messages and option selection facilitates. Emphasis is given for speedy processing and speedy transaction between the screens. Each screen assigned to make it as much user friendly as possible by

using interactive procedures. So to say user can operate the system without much help from the operating manual.

IMPLEMENTATION

OVERVIEW OF SOFTWARE TOOLS:-

HTML

Html is a language which is used to create web pages with html marking up a page to indicate its format, telling the web browser where you want a new line to begin or how you want text or images aligned and more are possible. The following tags in the project.

TABLE:

Tables are so popular with web page authors is that they let you arrange the elements of a web page in such a way that the browser won't rearrange them web page authors frequently use tables to structure web pages.

<TR>:

<TR> is used to create a row in a table encloses <TH> and <TD> elements. <TR> contain many attributes. Some of them are,

- ALIGN: specifies the horizontal alignment of the text in the table row.
- BGCOLOR: Specifies the background color for the row.
- BORDERCOLOR: Sets the external border color for the row.
- VALIGN: Sets the vertical alignment of the data in this row.

<TH>:

<TH> is used to create table heading.

- ALIGN: Sets the horizontal alignment of the content in the table cell. Sets LEFT, RIGHT, CENTER.
- BACKGROUND: Species the back ground image for the table cell.
- BGCOLOR: Specifies the background color of the table cell
- VALIGN: Sets the vertical alignment of the data. Sets to TOP, MIDDLE, BOTTOM or BASELINE.

- WIDTH: Specifies the width of the cell. Set to a pixel width or a percentage of the display area.

<TD>:

<TD> is used to create table data that appears in the cells of a table.

- ALIGN: Species the horizontal alignment of content in the table cell. Sets to LEFT, CENTER, RIGHT.
- BGCOLOR: Specifies the background image for the table cell.
- BGCOLOR: sets the background color of the table cells.
- WIDTH: Species the width of the cell

FRAMES:

Frames are used to either run off the page or display only small slices of what are supposed to be shown and to configure the frame we can use <FRAMESET> There are two important points to consider when working with <FRAMESET>.

- <FRAMESET> element actually takes the place of the <BODY> element in a document.
- Specifying actual pixel dimensions for frames .

<FRAME> Elements are used to create actual frames. From the frameset point of view dividing the browser into two vertical frames means creating two columns using the <FRAMESET> elements COLS attribute.

The syntax for vertical fragmentation is,

```
<FRAMESET COLS ="50%, 50%">
```

```
</FRAMESET>
```

Similarly if we replace COLS with ROWS then we get horizontal fragmentation.

The syntax for horizontal fragmentation is,

```
<FRAMESET ROWS ="50%, 50%">
```

```
</FRAMESET>
```

FORM:

The purpose of FORM is to create an HTML form; used to enclose HTML controls, like buttons and text fields.

ATTRIBUTES:

- ACTION: Gives the URL that will handle the form data.
- NAME: Gives the name to the form so you can reference it in code set to an alphanumeric string.
- METHOD: method or protocol is used to sending data to the target action URL. The GET method is the default, it is used to send all form name/value pair information in an URL. Using the POST method, the content of the form are encoded as with the GET method, but are sent in environment variables.

CONTROLS IN HTML

□ **<INPUT TYPE =BUTTON>:**

Creates a html button in a form.

ATTRIBUTES:

- NAME: gives the element a name. Set to alphanumeric characters.
- SIZE: sets the size.
- VALUE: sets the caption of the element.

□ **<INPUT TYPE = PASSWORD>:**

Creates a password text field, which makes typed input.

ATTRIBUTES:

- NAME: gives the element a name, set to alphanumeric characters.
- VALUE: sets the default content of the element.

□ **<INPUT TYPE=RADIO>:**

Creates a radio button in a form.

ATTRIBUTE:

- NAME: Gives the element a name. Set to alphanumeric character.
- VALUE: Sets the default content of the element.

□ **<INPUT TYPE=SUBMIT>:**

Creates a submit button that the user can click to send data in the form back to the web server.

ATTRIBUTES:

NAME: Gives the element a name. Set to alphanumeric characters.

VALUE: Gives this button another label besides the default, Submit Query. Set to alphanumeric characters.

<INPUT TYPE=TEXT>:

Creates a text field that the user can enter or edit text in.

ATTRIBUTES:

NAME: Gives the element a name. Set to alphanumeric characters.

VALUE: Holds the initial text in the text field. Set to alphanumeric characters.

JAVA Script

Java script originally supported by Netscape navigator is the most popular web scripting language today. Java script lets you embedded programs right in your web pages and run these programs using the web browser. You place these programs in a <SCRIPT> element, usually with in the <HEAD> element. If you want the script to write directly to the web page, place it in the <BODY> element.

JAVASCRIPT METHODS:

□ **WriteIn:**

Document.writeln() is a method, which is used to write some text to the current web page.

□ **onClick:**

Occurs when an element is clicked.

□ **onLoad:**

Occurs when the page loads.

□ **onMouseDown:**

Occurs when a mouse button goes down.

□ **onMouseMove:**

Occurs when the mouse is moved.

□ **onUnload:**

Occurs when a page is unloaded.

ORACLE

Oracle is a relational database management system, which organizes data in the form of tables. Oracle is one of many database servers based on RDBMS model, which manages a sea of data that attends three specific things-data structures, data integrity and data manipulation. With oracle cooperative server technology we can realize the benefits of open, relational systems for all the applications. Oracle makes efficient use of all systems resources, on all hardware architecture; to deliver unmatched performance, price performance and scalability. Any DBMS to be called as RDBMS has to satisfy Dr.E.F.Codd's rules.

DISTINCT FEATURES OF ORACLE:-

- **ORACLE IS PORTABLE:**

The Oracle RDBMS is available on wide range of platforms ranging from PCs to super computers and as a multi user loadable module for Novel NetWare, if you develop application on system you can run the same application on other systems without any modifications.

- **ORACLE IS COMPATIBLE:**

Oracle commands can be used for communicating with IBM DB2 mainframe RDBMS that is different from Oracle, that is Oracle compatible with DB2. Oracle RDBMS is a high performance fault tolerant DBMS, which is specially designed for online transaction processing and for handling large database applications.

- **MULTITHREADED SERVER ARCHITECTURE:**

Oracle adaptable multithreaded server architecture delivers scalable high performance for very large number of users on all hardware architecture including symmetric multiprocessors (sumps) and loosely coupled multiprocessors. Performance is achieved by eliminating CPU, I/O, memory and operating system bottlenecks and by optimizing the Oracle DBMS server code to eliminate all internal bottlenecks.

FEATURES OF ORACLE:-

Most popular RDBMS in the market because of its ease of use

- Client/server architecture.
- Data independence.
- Ensuring data integrity and data security.
- Managing data concurrency.
- Parallel processing support for speed up data entry and online transaction processing used for applications.
- DB procedures, functions and packages.

JDBC DRIVERS:

Type2 drivers are written in mixture of java and native code. Type2 drivers use vendors specific native APIs for accessing the data source. These drivers transform the JDBC calls to vendor specific calls using the vendor's native library. These drivers are also not portable like type1 drivers because of the dependency on native code.

TYPE3:

Type3 drivers use an intermediate middleware server for accessing the external data sources. The calls to the middleware server are database independent. However, the middleware server makes vendor specific native calls for accessing the data source. In this case, the driver is purely written in java.

TYPE4:

Type4 drivers are written in pure java and implement the JDBC interfaces and translate the JDBC specific calls to vendor specific access calls. They implement the data transfer and network protocol for the target resource manager. Most of the leading database vendors provide type4 drivers for accessing their database servers.

DRIVER MANAGER AND DRIVER:

The java.sql package defines an interface called Java.sql.Driver that makes to be implemented by all the JDBC drivers and a class called java.sql.DriverManager that acts as the interface to the database clients for performing tasks like connecting to external resource managers, and setting log streams. When a JDBC client requests the DriverManager to make a connection to an external resource manager, it delegates the task to an appropriate driver class implemented by the JDBC driver provided either by the resource manager vendor or a third party.

JAVA.SQL.DRIVERMANAGER:

The primary task of the class driver manager is to manage the various JDBC drivers register. It also provides methods for:

- Getting connections to the databases.
- Managing JDBC logs.
- Setting login timeout.

MANAGING DRIVERS:

JDBC clients specify the JDBC URL when they request a connection. The driver manager can find a driver that matches the request URL from the list of registered drivers and delegate the connection request to that driver if it finds a match. JDBC URLs normally take the following format:

<protocol>:<sub-protocol>:<resource>

The protocol is always jdbc and the sub-protocol and resource depend on the type of resource manager. The URL for PostgreSQL is in the format:

Jdbc: postgres ://< host> :< port>/<database>

Here host is the host address on which post master is running and database is the name of the database to which the client wishes to connect.

MANAGING CONNECTION:

DriverManager class is responsible for managing connections to the databases:

```
public static Connection getConnection (String url, Properties info) throws  
SQLException
```

This method gets a connection to the database by the specified JDBC URL using the specified username and password. This method throws an instance of SQLException if a database access error occurs.

CONNECTIONS:

The interface java.sql.Connection defines the methods required for a persistent connection to the database. The JDBC driver vendor implements this interface. A database 'vendor-neutral' client never uses the implementation class and will always use only the interface. This interface defines methods for the following tasks:

- Statements, prepared statements, and callable statements are the different types of statements for issuing SQL statements to the database by the JDBC clients.
- For getting and setting auto-commit mode.
- Getting meta information about the database.
- Committing and rolling back transactions.

CREATING STATEMENTS:

The interface `java.sql.Connection` defines a set of methods for creating database statements. Database statements are used for sending SQL statements to the database:

Public Statement `createStatement ()` throws `SQLException`

This method is used for creating instances of the interface `java.sql.Statement`. This interface can be used for sending SQL statements to the database. The interface `java.sql.Statement` is normally used for sending SQL statements that don't take any arguments. This method throws an instance of `SQLException` if a database access error occurs:

Public Statement `createStatement (int resType, int resConcurrency)` throws `SQLException`

JDBC RESULTSETS:

A JDBC resultset represents a two dimensional array of data produced as a result of executing SQL SELECT statements against databases using JDBC statements. JDBC resultsets are represented by the interface `java.sql.ResultSet`. The JDBC vendor provider provides the implementation class for this interface.

SCROLLING RESULTSETS:

public boolean `next()` throws `SQLException`

public boolean `previous()` throws `SQLException`

public boolean `first()` throws `SQLException`

public boolean `last()` throws `SQLException`

ACCESSING RESULTSET DATA:

Method name and Purpose	
public boolean <code>getBoolean (int i)</code>	
Gets the data in the specified column as a boolean.	
public boolean <code>getBoolean (String col)</code>	
public int <code>getInt(int i)</code>	Gets the data in the specified column as an int.
public int <code>getInt (String col)</code>	
public String <code>getString (int i)</code>	Gets the data in the specified column as a string.

STATEMENT:

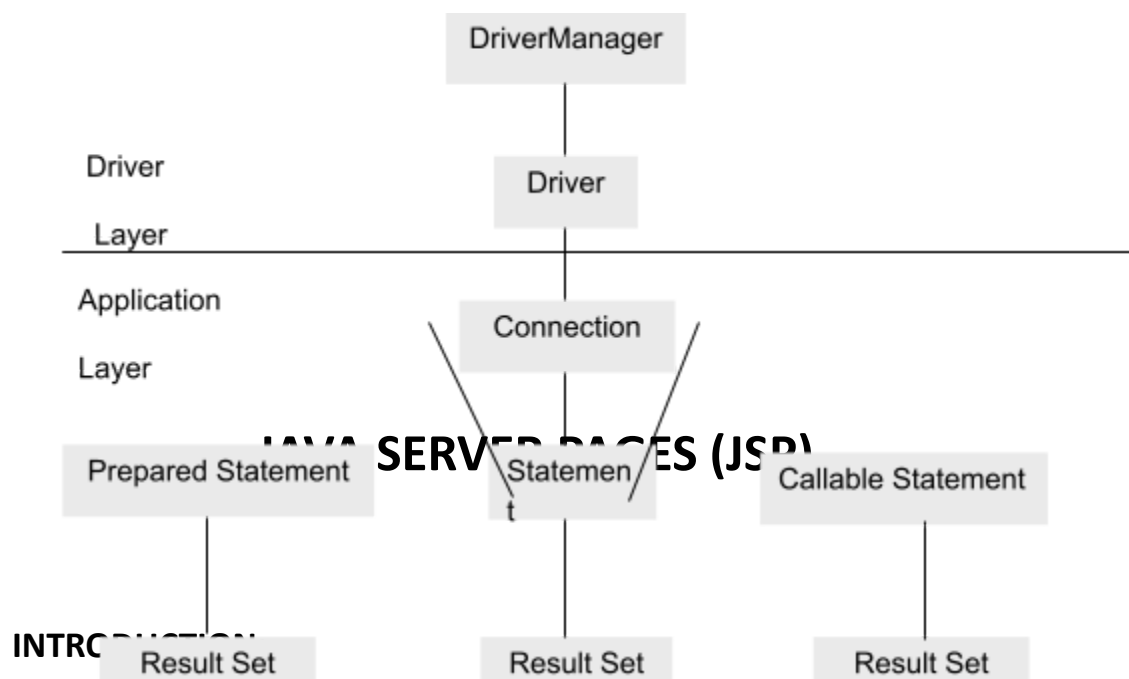
The interface `java.sql.Statement` is normally used for sending SQL statements that do not have IN or OUT parameters. The JDBC driver vendor provides the implementation class for this interface. The common methods required by the different JDBC statements are defined in this interface. The methods defined by `java.sql.Statement` can be broadly categorized as follows:

- Executing SQL statements
- Querying results and resultsets
- Handling SQL batches
- Other miscellaneous methods

The interface `java.sql.Statement` defines methods for executing different SQL statements like SELECT, UPDATE, INSERT, DELETE, and CREATE.

❑ `public ResultSet executeQuery(String sql)` throws `SQLException`

The following figure shows how the `DriverManager`, `Driver`, `Connection`, `Statement`, `ResultSet` classes are connected.



Java Server Pages (JSP) technology enables you to mix regular, static HTML with dynamically generated content. You simply write the regular HTML in the normal manner, using familiar Web-page-building tools. You then enclose the code for the dynamic parts in special tags, most of which start with `<%` and end with `%>`.

THE NEED FOR JSP:

Servlets are indeed useful, and JSP by no means makes them obsolete. However,

- It is hard to write and maintain the HTML.
- You cannot use standard HTML tools.
- The HTML is inaccessible to non-Java developers.

BENEFITS OF JSP:

JSP provides the following benefits over servlets alone:

- It is easier to write and maintain the HTML: In this no extra backslashes, no double quotes, and no lurking Java syntax.
- You can use standard Web-site development tools:

We use Macromedia Dreamweaver for most of the JSP pages. Even HTML tools that know nothing about JSP can be used because they simply ignore the JSP tags.

- You can divide up your development team:

The Java programmers can work on the dynamic code. The Web developers can concatenate on the representation layer. On large projects, this division is very important. Depending on the size of your team and the complexity of your project, you can enforce a weaker or stronger separation between the static HTML and the dynamic content.

CREATING TEMPLATE TEXT:

A large percentage of our JSP document consists of static text known as template text. In almost all respects, this HTML looks just like normal HTML follows all the same syntax rules, and simply “passed through” to that client by the servlet created to handle the page. Not only does the HTML look normal, it can be created by whatever tools you already are using for building Web pages.

There are two minor exceptions to the “template text passed through” rule. First, if you want to have `<% Or %>` in the out port, you need to put `<\% or %\>` in the template text. Second, if you want a comment to appear in the JSP page but not in the resultant document,

`<%-- JSP Comment -- %>`

HTML comments of the form:

`<!--HTML Comment -->`

are passed through to the client normally.

TYPES OF JSP SCRIPTING ELEMENTS:

JSP scripting elements allow you to insert Java code into the servlet that will be generated from the JSP page. There are three forms:

1. **Expressions** of the form `<%=Java Expression %>`, which are evaluated and inserted into the servlet’s output.
2. **Scriptlets** of the form `<%Java code %>`, which are inserted into the servlet’s `_jspService` method (called by service).
3. **Declarations** of the form `<%! Field/Method Declaration %>`, which are inserted into the body of the servlet class, outside any existing methods.

USING JSP EXPRESSIONS:

A JSP element is used to insert values directly into the output. It has the following form:

`<%= Java Expression %>`

The expression is evaluated, converted to a string, and inserted in the page. This evaluation is performed at runtime (when the page is requested) and thus has full access to the information about the request. For example, the following shows the date/time that the page was requested.

Current time: `<%=new java.util.Date () %>`

PREDEFINED VARIABLES

To simplify expressions we can use a number of predefined variables (or “implicit objects”). The specialty of these variables is that, the system simply tells what names it will use for the local variables in `_jspService`. The most important ones of these are:

- **Request** , the `HttpServletRequest`.
- **Response** , the `HttpServletResponse`.
- **Session** , the `HttpSession` associated with the request
- **Out** , the writer used to send output to clients.
- **Application** , the `ServletContext`. This is a data structure shared by all servlets and JSP pages in the web application and is good for storing shared data.

Here is an example: Your hostname: `<%= request.getRemoteHost () %>`

COMPARING SERVLETS TO JSP PAGES

JSP works best when the structure of the HTML page is fixed but the values at various places need to be computed dynamically. If the structure of the page is dynamic, JSP is less beneficial. Some times servlets are better in such a case. If the page consists of binary data or has little static content, servlets are clearly superior. Sometimes the answer is neither servlets nor JSP alone, but rather a combination of both.

WRITING SCRIPTLETS:

If you want to do something more complex than output the value of a simple expression .JSP scriptlets let you insert arbitrary code into the servlet's `_jspService` method. Scriptlets have the following form:

```
<% Java code %>
```

Scriptlets have access to the same automatically defined variables as do expressions (request, response, session, out , etc) .So for example you want to explicitly send output of the resultant page , you could use the out variable , as in the following example:

```
<%
```

```
String queryData = request.getQueryString ();  
  
out.println ("Attached GET data: "+ queryData);  
  
%>
```

SCRIPTLET EXAMPLE:

As an example of code that is too complex for a JSP expression alone, a JSP page that uses the bgColor request parameter to set the background color of the page .Simply using

```
<BODY BGCOLOR="<%= request.getParameter ("bgcolor") %"> ">
```

would violate the cardinal rule of reading form data.

DECLARATION:

A JSP declaration lets you define methods or fields that get inserted into the main body of the servlet class .A declaration has the following form:

```
<%! Field or Method Definition %>
```

Since declarations do not generate output, they are normally used in conjunction with JSP expressions or scriptlets. In principle, JSP declarations can contain field (instance variable) definitions, method definitions, inner class definitions, or even static initializer blocks: anything that is legal to put inside a class definition but outside any existing methods. In practice declarations almost always contain field or method definitions.

We should not use JSP declarations to override the standard servlet life cycle methods. The servlet into which the JSP page gets translated already makes use of these methods. There is no need for declarations to gain access to service, doGet, or doPost, since calls to service are automatically dispatched to _jspService , which is where code resulting from expressions and scriptlets is put. However for initialization and cleanup, we can use jsplnit and jspDestroy- the standard init and destroy methods are guaranteed to call these methods in the servlets that come from JSP.

JAKARTA TOMCAT

Tomcat is the Servlet/JSP container. Tomcat implements the Servlet 2.4 and JavaServer Pages 2.0 specification. It also includes many additional features that make it a useful platform for developing and deploying web applications and web services.

TERMINOLOGY:

Context – Context is a web application.

\$CATALINA_HOME – This represents the root of Tomcat installation.

DIRECTORIES AND FILES:

/bin – Startup, shutdown, and other scripts. The *.sh files (for Unix systems) are functional duplicates of the *.bat files (for Windows systems). Since the Win32 command-line lacks certain functionality, there are some additional files in here.

/conf – Configuration files and related DTDs. The most important file here is server.xml. It is the main configuration file for the container.

/logs – Log files are here by default.

/webapps – This is where webapps go\

INSTALLATION:

Tomcat will operate under any Java Development Kit (JDK) environment, that provides a JDK 1.2 (also known as Java2 Standard Edition, or J2SE) or later platform. JDK is needed so that servlets, other classes, and JSP pages can be compiled.

DEPLOYMENT DIRECTORIES FOR DEFAULT WEB APPLICATION:-

HTML and JSP FILES

- Main Location

\$CATALINA_HOME/webapps/ROOT

- Corresponding URLs.

<http://host/SomeFile.html>

<http://host/SomeFile.jsp>

- More Specific Location (Arbitrary Subdirectory).

\$CATALINA_HOME/webapps/ROOT/SomeDirectory

- Corresponding URLs

<http://host/SomeDirectory/SomeFile.html>

<http://host/SomeDirectory/SomeFile.jsp>

Individual Servlet and Utility Class Files

- Main Location (Classes without Packages).

\$CATALINA_HOME/webapps/ROOT/WEB-INF/classes

- Corresponding URL (Servlets).

<http://host/servlet/ServletName>

- More Specific Location (Classes in Packages).

\$CATALINA_HOME/webapps/ROOT/WEB-INF/classes/packageName

- Corresponding URL (Servlets in Packages).

<http://host/servlet/packageName.ServletName>

Servlet and Utility Class Files Bundled in JAR Files

- Location

\$CATALINA_HOME/webapps/ROOT/WEB-INF/lib

- Corresponding URLs (Servlets)

<http://host/servlet/ServletName>, <http://host/servlet/packageName.ServletName>

TESTING

SOFTWARE TESTING:-

Testing;

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

TESTING OBJECTIVES:

- To ensure that during operation the system will perform as per specification.
- TO make sure that system meets the user requirements during operation
- To make sure that during the operation, incorrect input, processing and output will be detected
- To see that when correct inputs are fed to the system the outputs are correct
- To verify that the controls incorporated in the same system as intended
- Testing is a process of executing a program with the intent of finding an error
- A good test case is one that has a high probability of finding an as yet undiscovered error

The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error. Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.,

TEST CASE DESIGN:

White box testing

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independent paths in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

Black Box Testing

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels:

Unit Testing:

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

Integration Testing:

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

Validation Testing

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

System Testing

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves:

- Implementing the system in a simulated production environment and testing it.
- Introducing errors and testing for error handling.

TEST CASES

TEST CASE 1:

Test case for Login form:

When a user tries to login by submitting an incorrect ID or an incorrect Password then it displays an error message "NOT A VALID USER NAME".

Test case 1: Verifying Authentication.		Priority (H, L): High
Test Objective: For Verifying Authentication.		
Test Description: "User (admin/supplier) enters username and password and presses <u>submit</u> button", client program contacts with server, server contacts with the database, database checks for authentication and sends result as a valid user.		
Requirements Verified: Yes		
Test Environment: Apache tomcat server must be in running state, Database Should contain appropriate table and link must be established between server and client program.		
Test Setup/Pre-Conditions: Apache server should be in running state. Username and Password fields should be entered.		
Actions		Expected Results
The user will select Login button to access application.		"Service Found", ".Displays Homepage of Corresponding user.
Pass: Yes	Conditions pass: Yes	Fail: No
Problems / Issues: NIL		
Notes: Successfully Executed		

TEST CASE 2:

Test case for User Registration form:

When a user enters user id to register and ID already exists, then this result in displaying error message "USER ID ALREADY EXISTS".

Requirements Verified: Yes	
Test Environment: Apache tomcat server must be in running state, Database Should contain appropriate table and link must be established between server and client program.	
Test Setup/Pre-Conditions: Apache server should be in running state. Values should be entered in all the fields should be entered.	
Actions	Expected Results
The supplier enters all the details and clicks submit button to access application.	"Service Found", ".Displays supplier registration successful.
Pass: Yes	Conditions pass: Yes Fail: No
Problems / Issues: NIL	
Notes: Successfully Executed	

TEST CASE 3:

Test case for Change Password:When the old password does not match with the new password,then this results in displaying an error message as " OLD PASSWORD DOES NOT MATCH WITH THE NEW PASSWORD".

Test case for Forget Password: When a user forgets his password he is asked to enter Login name, ZIP code, Mobile number. If these are matched with the already stored ones then user will get his Original password.

TEST CASE 4:

Test case for modification: When the buyer want to modify the tender correctly we get a message "MODIFIED THE TENDER".

TEST CASE 5:

Validation cases :

If any data field which accept the data from the user are not filled then the corresponding error messages are generated.

SCREEN SHOTS



□ This is the home page for "Tendering System". It is home.html.

Supplier Registration - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Reload Print Mail W Word PDF Help People

Address http://localhost:8080/tendering/supplier_registration.html Go Links

Supplier Registration

User Name :

Password :

Confirm Password :

Company :

Address :

Mobile No.(+91):

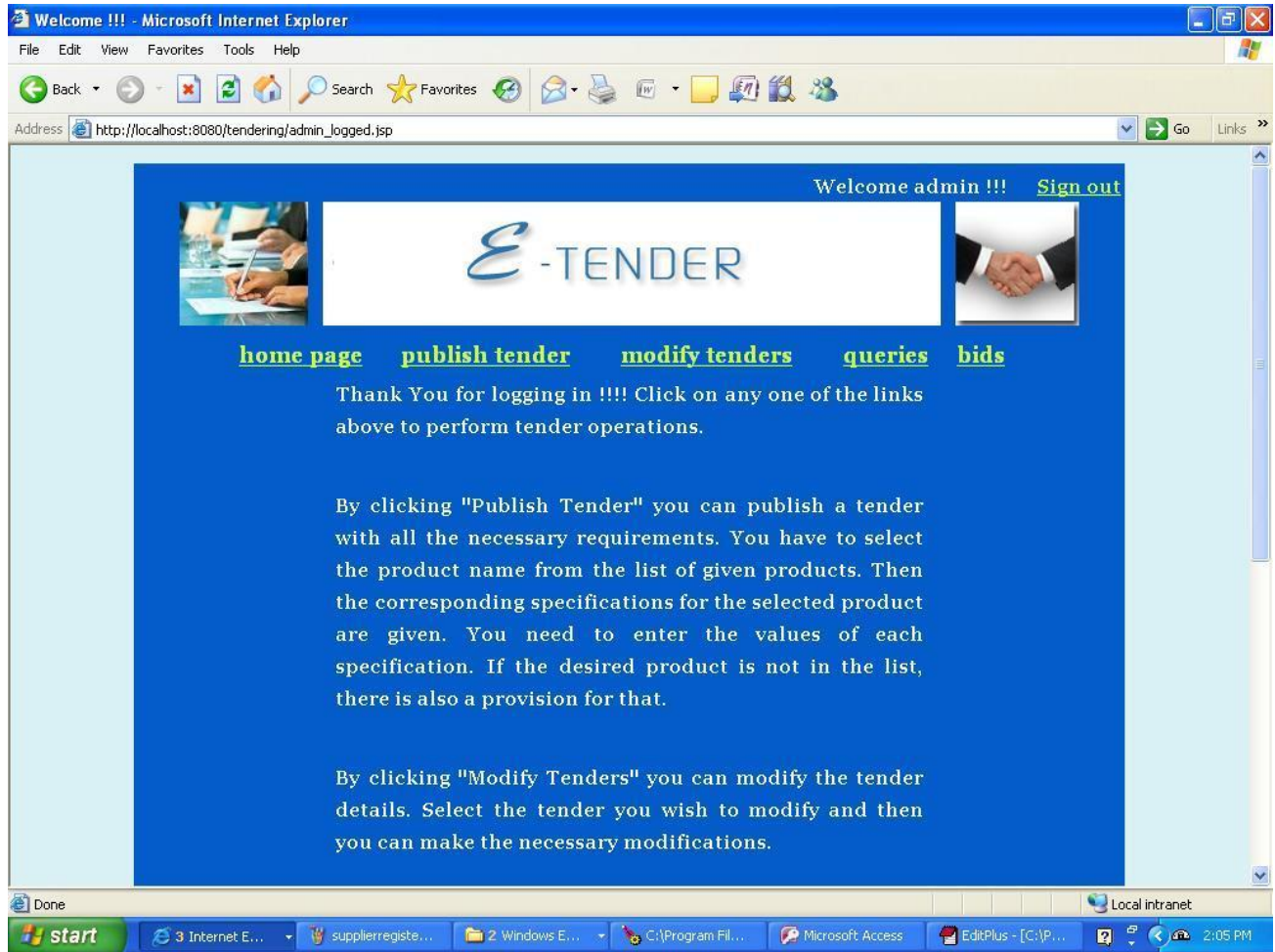

Copyright ©E-Corp.

Done Local intranet

start 3 Internet... Iviewpend... homepage... 2 Windo... C:\Progra... Microsoft A... EditPlus - [...]

2:04 PM

- If the supplier is new, then he can register in the website. If he clicks on “Register” link in home page, the above page is displayed.



- Once the admin logs into the website using his name and password, the above page is displayed. This is the home page for admin.

Tender Publishing Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites

Address http://localhost:8080/tendering/publish_tender.jsp?add=no Go Links

Welcome admin!!! [Sign out](#)

[home page](#) [publish tender](#) [modify tenders](#) [queries](#) [bids](#)

Tender Publishing

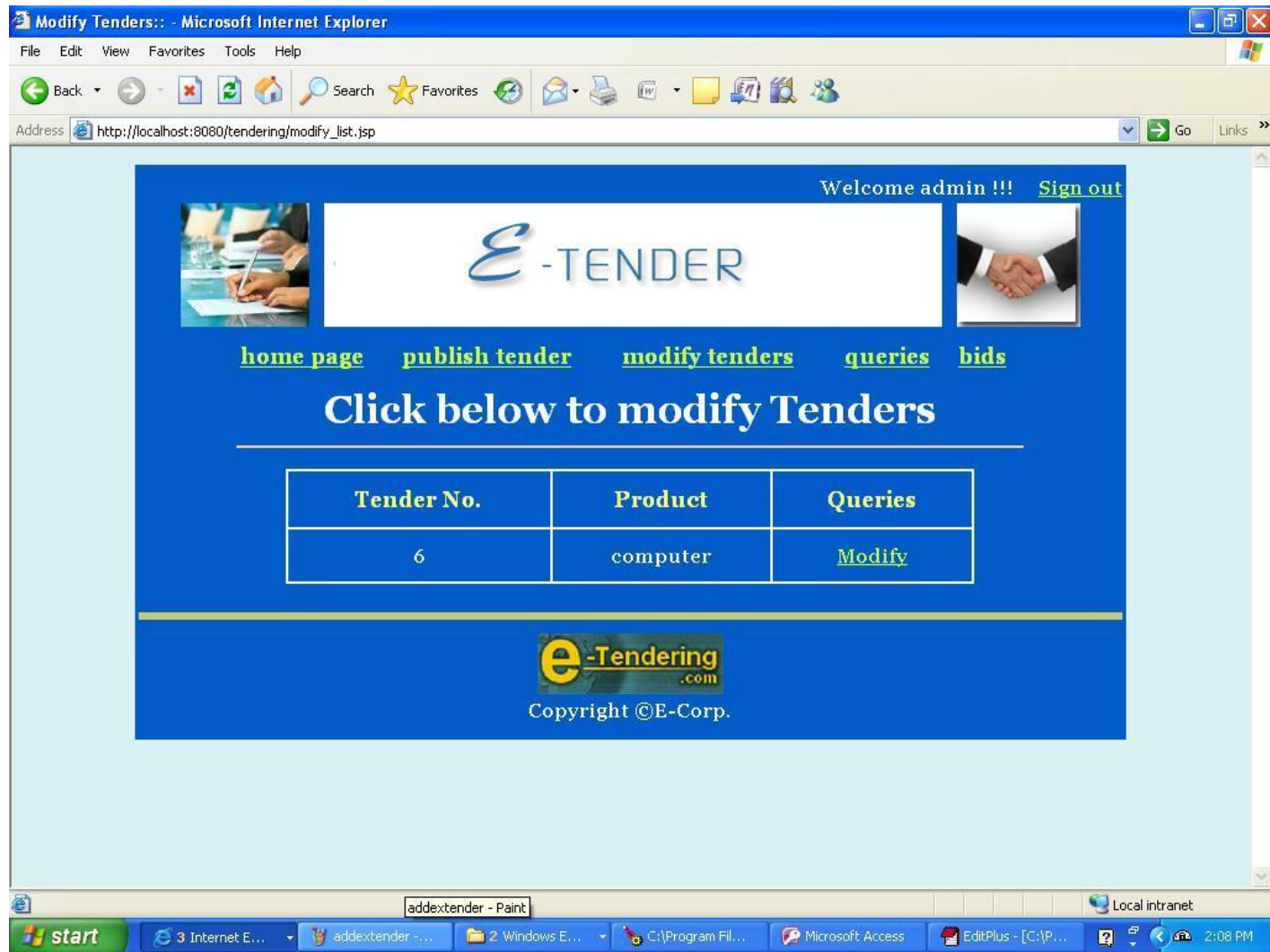
Select a product:

[Product not found?? Click here](#)

Tender No. : 7
Product Id: : 1
Quantity:
RAM : MB
Hard Disk : GB
Processor : GHz
Closing Date: :

start 3 Internet E... addtender - P... 2 Windows E... C:\Program Fil... Microsoft Access EditPlus - [C:\P... Local intranet 2:07 PM

- Once the product name and no. of specifications is given, the above page is displayed. Admin has to enter specification name and specification value for each specification and then click on “Submit”.



The above page is displayed when admin clicks on “Modify Tenders” link. All the list of open tenders will be displayed because admin can modify only for open tenders and not for closed tenders. To modify a particular product details, he has to click on any one of the links.

Bids - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites RSS Print Mail New Tab

Address <http://localhost:8080/tendering/bids.jsp> Go Links

Welcome admin !!! [Sign out](#)

[home page](#) [publish tender](#) [modify tenders](#) [queries](#) [bids](#)

Click below to view Bids

Tender No.	Product	Bids
6	computer	View Bids
5	tubelight	View Bids
4	tables	View Bids
3	tables	View Bids
2	tables	View Bids
1	White Board	View Bids

start 3 Internet E... view Q - Paint 2 Windows E... C:\Program Fil... Microsoft Access EditPlus - [C:\P... Local intranet 2:11 PM

The above page is displayed when admin clicks on “Bids” link. All the list of tenders appears and admin has to click on one of “View Bids” to view bids of supplier for a particular product.

The screenshot shows a Microsoft Internet Explorer window titled "Bid Form - Microsoft Internet Explorer". The address bar displays "http://localhost:8080/tendering/bid_form.jsp". The webpage has a blue background and contains the following elements:

- Top right: "Welcome abc!!!" and a [Sign out](#) link.
- Center: A large "E-TENDER" logo.
- Below the logo: Three links: [my homepage](#), [open tenders](#), and [closed tenders](#).
- Section title: "Bidding Form".
- Form fields:
 - "User : abc"
 - "Quoted Value(in Rs.) : " followed by a text input field.
 - "Any more Details : " followed by a larger text area.
 - A "Bid" button below the text area.
- Bottom: "e-Tendering.com" logo and "Copyright ©E-Corp."

The Windows taskbar at the bottom shows the Start button, several open applications (Internet Explorer, closedtenders, Windows Explorer, C:\Program Fil..., Microsoft Access, EditPlus), and the system clock showing 2:16 PM on a local intranet.

After viewing the tender details and product specifications, the supplier can bid for the product. The above page is displayed when the supplier clicks on "Bid for this product".

CONCLUSION

The E-Tendering project has been successfully completed. The goal of the system is achieved and the problems are solved. This project is developed in a manner that is user friendly and required help is provided at different levels.

The primary objective is to provide the interactive service to all the users. Different types of services are provided to both the supplier and buyer. This system certainly provides a better interface which can be helpful for both the supplier and buyer to make their working smooth.

LIMITATIONS

The present system is developed takes into account that the tenders posted are genuine in validation is done with respect to this. The system also does not verify the buyer's identification. As the memory capacity is limited to server's memory we can't accommodate infinite number of users.

FUTURE ENHANCEMENTS

In future the buyer's identity can be verified. In future the suppliers past supply details can be stored which the buyers can view to decide whether to give them the tender or not.

BIBILIOGRAPHY

Advanced Java Programming	- Dietel and Dietel
Mastering JAVA 2	- John Zukowski
Java Server Programming	- Apress

Software Engineering

- Roger S Pressman

Analysis & Design of Information Systems – Senn

Websites

www.eci.gov.in

www.google.com

www.apeci.com

www.askjeeves.com