Etransactions

ABSTRACT

E-Business Mart is a leading B2B marketplace that assists buyers and sellers to trade with each other at a common, reliable & transparent platform. The project is a web portal providing a marketplace to generate business leads, find suppliers, post trade offers, test market products, promote their brand and conduct business.

Project helps business people providing online catalogs, active products, business enhancements, mailing and communication between business people, b2b market places, trade enquiries, trade shows, online new business management, online survey for products. In addition to business directory, online catalog, trade leads, we offer personalized assistance to buyers and sellers for almost everything they need to buy/sell using mart process. Business people can perform B2B marketplace through the country or world.

Through the portal buyers, sellers, manufacturing companies, dealers, agencies and other business people will be provided a wide market for improving their business contacts and finding the best products and prices. The project also gives guidelines to customers to know various details of product catalogues, companies and business information.

The project provides an interaction between the business people around the country. It helps businessman in providing information of any product their required.

**INTRODUCTION**

This project involves the development of a back-end and front-end for the   
web-based business information. The basic goal of the project is able to produce a Web-Based System (Using a 3-Tier Architecture) that uses efficient server-side Web technologies, is platform independent and flexible.

The project entitled “**e-Business Mart**” provides entire information of a business online. **e-Business Mart**  project provides information related to products, business, industries, factories and various other information related to business. The project gives many benefits to buyers and sellers.

Project helps business people providing online catalogs, active products, business enhancements, mailing and communication between business people, b2b market places, trade enquiries, trade shows, online new business management, online survey for products . In addition to business directory, online catalog, trade leads and, we offer personalized assistance to buyers and sellers for almost everything they need to buy/sell using mart process. Business people can perform B2B marketplace through the country or world.

The project provides a interaction between the business people around the country. It helps businessman in providing information of any product their required.

PROJECT SYNOPSIS

1. SYNOPSIS

The project entitled “**e-Business Mart**” provides entire information of a business online. **e-Business Mart**  project provides information related to products, business, industries, factories, and various other information related to business. The project gives many benefits to buyers and sellers.

**OBJECTIVES OF THE PROJECT:**

# BENEFITS TO BUYERS

* Finding manufacturers and suppliers of desired products
* Locating manufacturers for customized product development/manufacturing
* Identifying new products for their market
* Seeking offers from sellers for your immediate/long-term buying requirements
* Establishing long-term association with companies for distribution, OEM supplies, contract manufacturing etc.
* Selecting the right service provider for various business needs and outsourcing of service requirements

# BENEFITS TO SELLERS

* Finding global buyers for their products and services
* Test marketing of new products globally for as low as Rs 12,000
* Setup a 24x7, 365 days online catalog for as low as Rs 19,000
* Invite interested buyers to buy their products through trade leads
* Brand building & generating potential sales leads through top listings, search listings and advertisements

**NEED FOR COMPUTERIZATION:**

* Reduces the work of officials in doing work manually for hours.
* Difficult to maintain information in manual process

2. PROBLEM DEFINITION AND EXISTING SYSTEM

The project entitled “**e-Business Mart**” provides entire information of a business online. **e-Business Mart**  project provides information related to products, business, industries, factories, and various other information related to business. The project gives many benefits to buyers and sellers.

The existed system is b2b is through business weekly magazines, news papers advertisements and marketing executives marketing. The existing is costly and covers only certain regions because giving advertisements, appointing marketing executives and processing the business in costly factor. The businessmen cannot also find competitive business market so the buyers and sellers will be in a static process.

3. PROPOSED SYSTEM

In the proposed project e-Business Mart interacts buyers and sellers to trade with each other at a common, reliable & transparent platform. The project is a web portal providing a marketplace to generate business leads, find suppliers, post trade offers, test market, business news, information, mail communication, trade enquiries, export and import companies information, products information, company information etc.,

We design a reliable and transparent platform for providing all the facilities. When the web portal is designed, businessmen all over country will have fast and best interaction and communication.

4. HARDWARE AND SOFTWARE SPECIFICATION

**HARDWARE SPECIFICATIONS**

Processor : Intel Pentium or more, IBM Cyrix (Intel compatible)

Hard Disk : 4.3 GB hard disk recommended for Primary partition.

Ram : Minimum 32 or 64 MB ram recommended for data

processing.

SOFTWARE SPECIFICATIONS

Operating System : Microsoft Windows 98/2000/XP

Front End : HTML, DHTML, JSP

Back End : Oracle

Web server : Apache Tomcat

Web browsers : Netscape Navigator or Internet Explorer

Scripting language : Java Script

Database drivers : Jdbc-Odbc Driver

FEASIBILITY STUDY

All projects are feasible – given unlimited resources and infinite time! Unfortunately, the development of computer-based system or product is more likely plagued by a scarcity of resources and difficult delivery dates. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. Months or years of effort, thousands or millions of dollars, and untold professional embarrassment can be averted if an ill-conceived system is recognized early in the definition phase.

**1. TECHNICAL FEASIBILITY:**

Technical to design this project is feasibly, the entire modules described in the modules description can be created using Front-End interaction JSP and back end database ORACLE.

**Advantages of ORACLE**

1. Oracle is a large database and several functional programs.
2. Oracle is pure database software. In our project we maintain database, so we selected Oracle
3. It provides a set of functional programs that user can use as tools to build structures and perform tasks.
4. Oracle is highly sucured software.
5. Oracle contains many tools like SQL, PL/SQL etc
6. SQL is a unified non-procedural language.

**2. ECONOMIC FEASIBILITY:**

In our project to find the cost of our project the following things are needed

Number of the persons(N)=3

Effort in person cost(E)=2000

Time period(t)=3 moths

Interface of the project(I)=10% of E

Number of modules(M)=2

Software cost(SC)=10000\*5%(including OS, Front end,Back end)

Project Cost=((N\*E)+(I\*M)+SC)\*t

=((3\*2000)+((2000\*0.1)\*2)+(10,000\*0.05)\*3

=(6000+400+500)\*3

=6900\*3

=20,700/-

**3. OPERATIONAL FEASIBILITY:**

In our application front end is developed using GUI. So it is very easy to the user to enter the necessary information. But user has some knowledge on using web applications before going to use our application.

CHAPTER - 1

PROJECT ANALYSIS

PROJECT ANALYSIS

1. 1. EXPLAIN ABOUT OUR ANALYZED CONCEPTS

Object oriented analysis is a method of analysis that examines the requirements from perspective of the classes and objects found in vocabulary of problem domain.

Analysis is the process of extracting the needs of a system and what that system must do to satisfy the client’s requirements.

**DOMAIN ANALYSIS**

Domain analysis is the process by which a software engineer learns background information, which helps to understand the problem. The word ‘domain’ in the case means the general field of business or technology in which the customers expect to be using the software.

For this project, personal experience of the team members and competing software of India Mart.com was observed to understand the domain.

**REQUIREMENTS ANALYSIS**

A requirement is a relatively short and concise piece of information, expressed as a fact. It can be written as a sentence or can be expressed using some kind of diagram.

Requirements are divided into two major types functional and non-functional.

**Functional Requirements**

Functional requirements describe what the system should do. The functional requirements can be further categorized as follows:

* What inputs the system should accept.
* What outputs the system should produce.
* What data the system must store.
* What are the computations to be done.
* The timing and synchronization of above.

**1.2. DATA FLOW DIAGRAMS**

A data flow diagram is graphical tool used to describe and analyze movement of data through a system. These are the central tool and the basis from which the other components are developed. The transformation of data from input to output, through processed, may be described logically and independently of physical components associated with the system. These are known as the logical data flow diagrams. The physical data flow diagrams show the actual implements and movement of data between people, departments and workstations. A full description of a system actually consists of a set of data flow diagrams. Using two familiar notations Yourdon, Gane and Sarson notation develops the data flow diagrams. Each component in a DFD is labeled with a descriptive name. Process is further identified with a number that will be used for identification purpose. The development of DFD’S is done in several levels. Each process in lower level diagrams can be broken down into a more detailed DFD in the next level. The lop-level diagram is often called context diagram. It consist a single process bit, which plays vital role in studying the current system. The process in the context level diagram is exploded into other process at the first level DFD.

**DFD NOTATIONS**

#### Define source and destination data.

#### Shows path of the data flow.

To represent a process that transforms or modifies the Data.

To represent an attribute.

#### Data Store

### **DATA FLOW DIAGRAMS**



**Fig.1. Data Flow Diagram for E-Businesses Mart**

1.3. CLASS DIAGRAMS



**Fig. 2. Overall Class Diagram of Business Mart**

**1.4. USE CASE DIAGRAMS**



**Fig. 3. Use Case Diagram for Admin**



**Fig. 4. Use Case Diagram for Users**

**1.5. SEQUENCE DIAGRAMS**



**Fig. 5. Sequence Diagram for Administrator**

**:user**

**:login**

**check**

**:view**

**:businessmen**

**1.enter**

**username**

**and password**

**2.check**

**details**

**3.provides**

**screen**

**4.view**

**business information and**

**process trade**

**enquriy**

**5.mail**

**to the administrator to clarify doubts**

**6.clear**

**user doubts and provide requested information**

**SEQUENCE DIAGRAM FOR USER**

Buy SmartDra

!

-

purchased copies print this

.

Visit www

.

.

com or call

1

-

800

-

768

-

39

.

**Fig: 6. Sequence Diagram for User Activities**

**1.6. COLLABORATION DIAGRAMS**



**Fig. 7. Collaboration Diagram for Admin**

# 



**Fig. 8. Collaboration Diagram for Business Men**



**Fig. 9. Collaboration Diagram for User**

1.7. DATA DICTIONARY

Database design is defined as the design of how data is persistently stored so that it may be accessed by many programs and users, over an indefinite period of time.

**TABLES**

|  |  |  |
| --- | --- | --- |
| **TABLE: ADMIN\_INBOX** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| MID | NUMBER | PRIMARY KEY |
| USER\_NAME | VARCHAR2(30) |  |
| MSG\_FROM | VARCHAR2(50) |  |
| MSG\_SUBJECT | VARCHAR2(50) |  |
| MSG\_DATA | VARCHAR2(1000) |  |
| MSG\_DATE | DATE |  |
|  |  |  |
| **TABLE: DIRECTORY** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| COMPANY | VARCHAR2(100) | PRIMARY KEY |
| ADDRESS | VARCHAR2(500) |  |
| COUNTRY | VARCHAR2(100) |  |
| STATE | VARCHAR2(100) |  |
| CITY | VARCHAR2(100) |  |
| PINCODE | NUMBER(10) |  |
| PHONE | NUMBER(15) |  |
| MOBILE | NUMBER(15) |  |
| EMAIL | VARCHAR2(100) |  |
| WEBSITE | VARCHAR2(100) |  |
| BPROFILE | VARCHAR2(500) |  |
| PPROFILE | VARCHAR2(500) |  |
|  |  |  |
| **TABLE: INDUSTRY** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| TYPE | VARCHAR2(100) |  |
| DETAILS | VARCHAR2(1000) |  |
|  |  |  |

|  |  |  |
| --- | --- | --- |
| **TABLE: PRODUCTS** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| COMPANY | VARCHAR2(100) |  |
| PRODUCTID | VARCHAR2(100) | PRIMARY KEY |
| NAME | VARCHAR2(150) |  |
| DETAILS | VARCHAR2(500) |  |
| IMAGE | VARCHAR2(100) |  |
| TYPEID | VARCHAR2(50) |  |
|  |  |  |
| **TABLE: PTYPE** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| TYPEID | VARCHAR2(50) | PRIMARY KEY |
| TYPE | VARCHAR2(100) |  |
|  |  |  |
|  |  |  |
| **TABLE: BIDS** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| BIDNO | NUMBER(10) | PRIMARY KEY |
| TID | NUMBER(10) |  |
| CID | VARCHAR2(10) |  |
| DOS | DATE |  |
| EMD | NUMBER(10) |  |
| DDNO | VARCHAR2(20) |  |
| NAMEOFBANK | VARCHAR2(20) |  |
| DODD | DATE |  |
| FEE | NUMBER(10,2) |  |
| DDFENO | VARCHAR2(20) |  |
| DOFD | DATE |  |
| NAMEOFBANK1 | VARCHAR2(20) |  |
| ECURITY | NUMBER(10) |  |
| BIDAMTPER | NUMBER(10,2) |  |
| STATUS | CHAR(1) |  |

|  |  |  |
| --- | --- | --- |
| **TABLE: ALLOTMENT** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| BIDNO | NUMBER(10) |  |
| DOA | DATE |  |
| CID | VARCHAR2(10) |  |
| AUTHORITY | VARCHAR2(20) |  |
|  |  |  |
| **TABLE: LIST** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| TID | NUMBER(10) | PRIMARY KEY |
| DOT | DATE |  |
| TOT | VARCHAR2(20) |  |
| TCAT | VARCHAR2(20) |  |
| BSD | DATE |  |
| BSSCD | DATE |  |
| DON | DATE |  |
| FEE | NUMBER(10) |  |
| EMD | NUMBER(10) |  |
| CID | VARCHAR2(20) |  |
| WORK | VARCHAR2(200) |  |
| ECOST | NUMBER(10) |  |
| POC | NUMBER(4) |  |
| BTYPE | VARCHAR2(30) |  |
| BCALL | NUMBER(2) |  |
| TOQ | VARCHAR2(30) |  |
| DOA | DATE |  |
| PAYMENTMODE | VARCHAR2(30) |  |
| PAYMENTPERENTAGE | NUMBER(10) |  |
| STATUS | CHAR(1) |  |

|  |  |  |
| --- | --- | --- |
| **TABLE: TRADESHOW** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| TRADEID | VARCHAR2(40) |  |
| FDATE | DATE |  |
| TDATE | DATE |  |
| TRADESHOW | VARCHAR2(100) |  |
| INDUSNAME | VARCHAR2(100) |  |
| EVENTPROFILE | VARCHAR2(1000) |  |
| VENUE | VARCHAR2(500) |  |
| CITY | VARCHAR2(100) |  |
| ORGANIZED | VARCHAR2(100) |  |
| MOBILE | NUMBER(15) |  |
|  |  |  |
| **TABLE: USERS** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| USERID | VARCHAR2(20) | PRIMARY KEY |
| PASSWORD | VARCHAR2(20) | PRIMARY KEY |
| TYPS | VARCHAR2(1) |  |
|  |  |  |
| **TABLE: USER\_INBOX** |  |  |
|  |  |  |
| **COLUMN NAME** | **DATATYPE** | **CONSTRAINTS** |
| MID | NUMBER | PRIMARY KEY |
| USER\_NAME | VARCHAR2(30) |  |
| MSG\_FROM | VARCHAR2(50) |  |
| MSG\_SUBJECT | VARCHAR2(50) |  |
| MSG\_DATA | VARCHAR2(500) |  |
| MSG\_DATE | DATE |  |

**1.8. SOFTWARE MODEL**

The software model used for our project is **‘WATERFALL MODEL’.**

**Waterfall Model:**

The general methodology in developing a system is involved in different phases, which describe the system’s life cycle model for developing software project. The concept includes not only forward motion but also have the possibility to return that is cycle back to an activity previously completed. This cycle back or feedback may occur as a result of the failure with the system to meet a performance objective or as a result of changes in redefinition of system activities. Like most systems, the life cycle of the computer-based system also exhibits distinct phases.

Those are,

1. Requirement analysis phase
2. Design phase
3. Development phase
4. Coding phase
5. Testing phase

Requirement Analysis Phase:

This phase includes the identification of the problem, in order to identify the problem, we have to know information about the problem, the purpose of the evaluation for problem to be known. We have to clearly know about the client’s requirements and the objectives of the project.

System Analysis Phase:

Feasibility analysis involves the benefits of various approaches and the determination of the alternative approaches although methods like questionnaires and interviews etc., different data about the project is collected and the data through out the project is represented in the form of UML Diagrams.

Design Phase:

S/W design is a process through which the requirements are translated into a representation of a s/w. The design of the system is in modular form i.e., the s/w is logically partitioned into components that perform specific functions and sub functions. The design phase leads to modules that exhibit independent functional characteristics. The design phase is of main importance because in this activity, decisions ultimately affect the success of s/w implementation and maintenance.

Development Phase:

The development phase includes choosing of a suitable s/w to solve the particular problem given. The various facilities and the sophistication in the selected s/w give a better development of the problem***.***

Coding Phase:

The coding phase is for translating the design of the system produced during the design phase into code in a given programming language, which can be executed by a computer and which performs the computation specified by the design.

Testing Phase:

# Testing is done in various ways such as testing the algorithm, programming code, sample data debugging is also one of following the above testing.

Requirement Analysis Phase

Design Phase

Deployement Phase

Coding Phase

Testing Phase

Fig. 10. Water Fall Model

CHAPTER - 2

PROJECT MODULES

PROJECT MODULES

The following are the modules of the project:

1. Admin Module
2. Businessmen Module
3. Trade shows, Directory Services & Catalogues Module
4. Module
5. Communications Module

ADMIN MODULE

The administrator is a person who looks after the entire project. He keeps track of all the actions performed various users in the system. The following are activities of administrator. Administrator provides information related to suppliers, post trade offers, test market, business news, information, mail communication, trade enquiries, export and import companies information, products information, company information etc.,

BUSINESSMEN MODULE

Businessmen module provides two categories one-supplier information and buyer information. The supplier may be industry, dealer or other category that has products for selling. The buyer may be a businessman who wants to purchase the goods. Each person is provides a registration for business entrepreneur to upload information of their products to website like products, rates, details, trade enquiry information and the businessmen can view all the information which is needed.

TRADE SHOWS, DIRECTORY SERVICES & CATALOGUES MODULE

DIRECTORY

Manufacturers Directory and Wholesale Business catalogs, an online B2B platform lets buyers interact with businesses of their interest, in domestic and international marketplace. Get access to an online directory of manufacturers and wholesalers, dealing in the manufacture and wholesale of a wide range of products and services. Portal provides various business director information like:

* + - * Business Director
      * Service Director
      * Supplier Director

TRADE LEAD

Project provides Trade Portal for international, import export trade lead and business trade leads generation and match-making platform. International buyers and sellers can post their trade leads to buy and sell, import and export a wide range of products and services.

TRADE SHOWS

Providing information of trade fairs and business events brings you an exhaustive coverage of exhibitions, trade shows & expositions, conferences and seminars for various industries worldwide. Businessman can browse through the most comprehensive information on individual trade events worldwide, along with their event profile, organizer, exhibitor and visitor profile, venues and dates to plan your participation much in advance.

MODULE

Providing information of various information being floated by government and private organizations all over India

### **COMMUNICATION MODULE**

Provides mail communication between businessmen, suppliers, companies for trade enquires.

CHAPTER – 3

CODING

CODING

**COMMON.JSP**

<%@ page import="java.sql.\*" %>

<%!

Connection con = null;

Statement st = null,st1=null,st2=null,st3=null;

%>

<%

Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");

con=DriverManager.getConnection("jdbc:odbc:emart","emart","emart");

st=con.createStatement();

st1=con.createStatement();

st2=con.createStatement();

st3=con.createStatement();

%>

**DIRECTORYADD.JSP**

<link href="ksrm.cdf" rel="stylesheet"></link>

<%@ page import="java.sql.\*" %>

<html>

<head>

<title> (products) Products Adding Form</title>

</head>

<script language="JavaScript">

function enableAll()

{

document.products.company.disabled=false;

document.products.address.disabled=false;

document.products.country.disabled=false;

document.products.state.disabled=false;

document.products.city.disabled=false;

document.products.pincode.disabled=false;

document.products.phone.disabled=false;

document.products.mobile.disabled=false;

document.products.email.disabled=false;

document.products.website.disabled=false;

document.products.bprofile.disabled=false;

document.products.pprofile.disabled=false;

document.products.company.focus();

}

function formvalid()

{

var company=document.products.company.value;

var address=document.products.address.value;

var country=document.products.country.value;

var state=document.products.state.value;

var city=document.products.city.value;

var pincode=document.products.pincode.value;

var phone=document.products.phone.value;

var mobile=document.products.mobile.value;

var email=document.products.email.value;

var website=document.products.website.value;

var bprofile=document.products.bprofile.value;

var pprofile=document.products.pprofile.value;

if((company.length==0) || (address.length==0) || (country.length==0) || (state.length==0) || (city.length==0)||(pincode.length==0) || (phone.length==0) || (mobile.length==0) || (email.length==0) || (website.length==0)|| (bprofile.length==0) || (pprofile.length==0))

{

alert('fill the form completely')

return false;

}

return true;

}

</script>

</head>

<body id=body >

<a href=admin.html><img src=images\p5.gif style="position:absolute;top:20;left:700;"></a>

<form name=products method=post action=directoryins.jsp >

<br>

<div>

<img src=images\p1.gif>

<TABLE WIDTH="90%" BORDER="1" CELLPADDING="0" CELLSPACING="0" BGCOLOR="#EFF8FF" BORDERCOLOR="#C2E6FE" STYLE="border-collapse:collapse;" ALIGN="CENTER" >

<TR>

<TD CLASS="glformhead" HEIGHT="24" VALIGN="MIDDLE" align=center><B>Directory Entry</B></TD></TR>

</table><TABLE WIDTH="90%" BORDER="1" CELLPADDING="8" CELLSPACING="0" BORDERCOLOR="#DDF0FF" STYLE="border-collapse:collapse;" ALIGN="CENTER">

<tr><td class=td><b>Company</b></td><td><input name=company type=text disabled size=60></td></tr>

<tr><td class=td><b>Address</b></td><td><input name=address type=text disabled size=60></td></tr>

<tr><td class=td><b>Country</b></td><td><input name=country type=text disabled size=60></td></tr>

<tr><td class=td><b>State</b></td><td><input name=state type=text disabled size=60></td></tr>

<tr><td class=td><b>City</b></td><td><input name=city type=text disabled size=60></td></tr>

<tr><td class=td><b>Pincode</b></td><td><input name=pincode type=text disabled size=60></td></tr>

<tr><td class=td><b>Phone</b></td><td><input name=phone type=text disabled size=60></td></tr>

<tr><td class=td><b>Mobile</b></td><td><input name=mobile type=text disabled size=60></td></tr>

<tr><td class=td><b>Email</b></td><td><input name=email type=text disabled size=60></td></tr>

<tr><td class=td><b>WebSite</b></td><td><input name=website type=text disabled size=60></td></tr>

<tr><td class=td><b>Business Profile</b></td><td><textarea rows=3 cols=45 name=bprofile disabled></textarea></td></tr>

<tr><td class=td><b>Product Profile</b></td><td><textarea rows=3 cols=45 name=pprofile disabled ></textarea></td></tr>

</table>

</div>

<input type="button" value="Add New" class=border onclick=enableAll();>

<input type="submit" value="Save" class=border></center>

</form>

<center>

</body>

</html>

**APPLY .JSP**

<link href="ksrm.cdf" rel="stylesheet">

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

<html>

<script language="JavaScript">

function dd()

{

bid.MonthView1.style.visibility = "Visible"

}

function dd1()

{

bid.MonthView2.style.visibility = "Visible"

}

document. bid.dofd.value = bid.MonthView2.day + "/" + mon + "/" + yer.substr(3,2);

bid.MonthView2.style.visibility = "Hidden";

}

function enableAll()

{

document. bid.cid.disabled=false;

document. bid.emd.disabled=false;

document. bid.ddno.disabled=false;

document. bid.nameofbank.disabled=false;

document. bid.dodd.disabled=false;

document. bid. fee.disabled=false;

document. bid.ddfeno.disabled=false;

document. bid.dofd.disabled=false;

document. bid.nameofbank1.disabled=false;

document. bid. ecurity.disabled=false;

document. bid.bidamtper.disabled=false;

}

function formvalid()

{

var emd=document. bid.emd.value;

var ddno=document. bid.ddno.value;

var nameofbank=document. bid.nameofbank.value;

var dodd=document. bid.dodd.value;

var fee=document. bid. fee.value;

var ddfeno=document. bid.ddfeno.value;

var dofd=document. bid.dofd.value;

var nameofbank1=document. bid.nameofbank1.value;

var ecurity=document. bid. ecurity.value;

var bidamtper=document. bid.bidamtper.value;

if((emd.length==0) || (ddno.length==0) || (nameofbank.length==0) || (dodd.length==0) || ( fee.length==0) || (ddfeno.length==0) || (dofd.length==0) || (nameofbank1.length==0) || ( ecurity.length==0) || (bidamtper.length==0) )

{

alert('fill the form completely')

return false;

}

return true;

}

</script>

<body>

<img src="http://localhost:8080/emart/images/p1.gif">

<div style="position:absolute;left:680;top:40;" >

<a href=company.html target=\_parent><img src="http://localhost:8080/emart/images/p5.gif"></a>

</div>

<form name= bid action= bidins.jsp onSubmit='return formvalid()'>

<OBJECT id=MonthView1 classid=clsid:8E27C92B-1264-101C-8A2F-040224009C02 style="position:absolute;top:250;left:500;VISIBILITY: hidden;" width=200 ></object>

<OBJECT id=MonthView2 classid=clsid:8E27C92B-1264-101C-8A2F-040224009C02 style="position:absolute;top:250;left:500;VISIBILITY: hidden;" width=200 ></object>

<%

int tid = Integer.parseInt(request.getParameter("tid"));

int emd = Integer.parseInt(request.getParameter("emd"));

int tfee = Integer.parseInt(request.getParameter("tfee"));

String cid = session.getValue("userid").toString();

%>

<center>

<TABLE WIDTH="90%" BORDER="1" CELLPADDING="0" CELLSPACING="0" BGCOLOR="#EFF8FF" BORDERCOLOR="#C2E6FE" STYLE="border-collapse:collapse;" ALIGN="CENTER" >

<TR>

<TD CLASS="glformhead" HEIGHT="24" VALIGN="MIDDLE" align=center><B>Apply Entry</B></TD></TR>

</table>

<TABLE WIDTH="90%" BORDER="1" CELLPADDING="8" CELLSPACING="0" BORDERCOLOR="#DDF0FF" STYLE="border-collapse:collapse;" ALIGN="CENTER">

<tr><td class=td><b> Id</b></td><td>

<input name=tid type=text size=10 readonly value=<%=tid%> onkeypress='if(event.keyCode<48 || event.keyCode>58){ bid.tid.focus(); return false;}if( bid.tid.value.length>10)return false;'></td></tr>

<tr><td class=td><b>Company Id</b></td><td>

<input name=cid type=text disabled size=10 readonly value=<%=cid%>></td></tr>

<tr><td class=td><b>Earnest Money Deposit</b></td><td>

<input name=emd type=text disabled size=10 readonly value=<%=emd%> onkeypress='if(event.keyCode<48 || event.keyCode>58){ bid.emd.focus(); return false;}if( bid.emd.value.length>10)return false;'></td></tr>

<tr><td class=td><b>DD No.</b></td><td>

<input name=ddno type=text disabled size=20 ></td></tr>

<tr><td class=td><b>Bank Name</b></td><td>

<input name=nameofbank type=text disabled size=20 ></td></tr>

<tr><td class=td><b>Date of DD</b></td><td>

<input name=dodd type=text disabled size=10 onfocus='dd();'><input type=button value=Calendar onclick='dd();'></td></tr>

<tr><td class=td><b> Fee</b></td><td>

<input name= fee type=text disabled size=10 readonly value=<%=tfee%> onkeypress='if(event.keyCode<48 || event.keyCode>58){ bid. fee.focus(); return false;}if( bid. fee.value.length>10)return false;'></td></tr>

<tr><td class=td><b>DD No.</b></td><td>

<input name=ddfeno type=text disabled size=20 ></td></tr>

<tr><td class=td><b>Date of DD</b></td><td>

<input name=dofd type=text disabled size=10 onfocus='dd2();'><input type=button value=Calendar onclick='dd1();'></td></tr>

<tr><td class=td><b>Name of Bank</b></td><td>

<input name=nameofbank1 type=text disabled size=20 ></td></tr>

<tr><td class=td><b> Security</b></td><td>

<input name= ecurity type=text disabled size=10 ></td></tr>

<tr><td class=td><b>Bid Amt</b></td><td>

<input name=bidamtper type=text disabled size=10 onkeypress='if(event.keyCode<48 || event.keyCode>58){ bid.bidamtper.focus(); return false;}if( bid.bidamtper.value.length>10)return false;'></td></tr>

</table>

<center>

<input type="button" value="Add New" class=border onclick=enableAll();>

<input type="submit" value="Save" class=border >

</form>

</fieldset>

</body>

</html>

CHAPTER – 4

SCREENS

SCREENS

CHAPTER - 5

TESTING AND DEBUGGING

TESTING AND DEBUGGING STRATEGIES

Object Oriented system architecture consists of any sub system that encapsulates collaboration classes. The sub system may communicate with other sub systems. A class performs certain operations. Collaborative classes perform one or few responsibilities. Thus more testing efforts are needed at different levels.

All Object Oriented models should be tested for correctness, completeness and consistency with in the model’s syntax, semantics and pragmatics.

**STEPS IN OBJECT ORIENTED TESTING**

1. All the class definitions and hierarchies for emissions and ambiguities.
2. Review and cross check the CRC model and object relationship model. Check all the responsibilities and collaborative.
3. Check whether you can club two or more responsibilities into one.
4. Check whether responsibilities are properly grouped.

**OBJECT ORIENTED TESTING STRATEGIES**

1. Unit Testing
2. Integration Testing
3. Validation and System Testing

**Unit Testing**

Unit Testing focuses verification efforts in smallest unit of software design. Unit Testing was performed on low-level modules, proceeding one at a time. Bpttom-up testing was performed on each module. After the lower level modules were tested, the modules that in the next higher level those make use of the lower modules were tested.

**Integration Testing**

Object Oriented software doesn’t have hierarchical structure. Hence top-down and bottom-up approaches have little meaning. Integrating one operation at a time into a class is also not possible. Hence the following strategies are used.

1. Thread-Based Testing
2. Use-Based Testing
3. Cluster-Based Testing

* **Thread–Based Testing**

Thread means a process. To identify various threads that is identifying each set of classes that respond to one input or measured in one output. Integrate each thread and perform testing. Regression testing is also done to ensure that there are no side effects.

* **Use-Based Testing**

All independent classes are tested first. Then test the dependent classes that use those dependent classes. This process is continued until the entire system is tested. Drivers and stubs are essential here. Driver is a program and stub is a sub program. They are avoided whenever possible.

* **Cluster-Based Testing**

A cluster of collaborating classes to uncover errors in collaborations. The clusters are determined by examining LRL model and object relational model.

**Validation and System Testing**

Like conventional testing, object oriented testing focuses on user visible action and user recognizable outputs from the system. For this testing we use usecases and object behavior model. We also use state transition diagram and event flow diagram in object oriented approach. In any case class is the target for testing.

Conventional blackbox methods can be used to draw validation tests. Requirement specification serves as the basis for validations.

**TESTING METHODS**

**White-Box Testing**

White-box testing is also called as glass box testing is attest case design method that uses the control structure of procedural design to derive test cases. It is ensured that all the independent paths within boundaries of all loops have been executed correctly and the validation of internal data structures is done.

Using this testing methods we can derive test cases that guarantee all independent paths with in a module has been exercised at least once, exercises all logical decisions on their false and true sides, executes all loops at their boundaries and within their operational bounds and exercise internal data a structures to assume their validity and termination.

**Condition Testing**

Condition testing is a test design method that executes logical conditions contained in program module. The condition testing focuses on each condition in the program.

**Black-Box Testing**

This focuses on the functional requirements of the software. It attempts to find errors in incorrect function, errors in data structures, interface errors, performance errors and initialization errors. During the implementation of user friendly a generic test is done.

It attempts to find errors in different categories as said above. By applying these techniques, we derive the set of test cases that satisfy the following criteria test cases that reduces by account greater than one, the number of additional test cases that tell us about the presence or absence of classes for errors, rather than errors associated only with the specific test at hand.

CHAPTER – 6

TECHNOLOGIES

ABOUT TECHNOLOGIES

**JAVA SERVER PAGES (JSP):**

Java server Pages is a simple, yet powerful technology for creating and maintaining dynamic-content web pages. Based on the Java programming language, Java Server Pages offer proven portability, open standards, and mature re-usable component model .The Java Server Pages architecture enables the separation of content generation from content presentation. This separation not only eases maintenance headaches but also allows web team members to focus on their areas of expertise. Now, web page designer can concentrate on layout, and web application designers on programming, with minimal concern about impacting each other work.

**Features of JSP:**

**Portability:**

Java Server Pages files can be run on any web server or web-enabled application server that provides support for them. Dubbed the JSP engine, this support involves recognition, translation, and management of the Java Server Page lifecycle and its interaction components.

**Components:**

It was mentioned earlier that the Java Server Pages architecture can include reusable Java components. The architecture also allows for the embedding of a scripting language directly into the Java Server Pages file. The components currently supported Java Beans, and Servlets.

**Processing:**

A Java Server Pages file is essentially an HTML document with JSP scripting or tags. The Java Server Pages file has a JSP extension to the server as a Java Server Pages file. Before the page is served, the Java Server Pages syntax is parsed and processed into a Servlet on the server side. The Servlet that is generated outputs real content in straight HTML for responding to the client.

**Access Models:**

A Java Server Pages file may be accessed in at least two different ways. A client’s request comes directly into a Java Server Page. In this scenario, suppose the page accesses reusable Java Bean components that performs particular well-defined computations like accessing a database. The result of the Beans computations, called result sets is stored within the Bean as properties. The page uses such Beans to generate dynamic content and present it back to the client.

In both of the above cases, the page could also contain any valid Java code. Java Server Pages architecture encourages separation of content from presentation.

**Steps in the execution of a JSP Application:**

1. The client sends a request to the web server for a JSP file by giving the name of the JSP file within the form tag of a HTML page.
2. This request is transferred to the JavaWebServer. At the server side JavaWebServer receives the request and if it is a request for a jsp file server it gives this request to the JSP engine.
3. JSP engine is program which can understand the tags of the jsp and then it converts those tags into a Servlet program and it is stored at the server side. This Servlet is loaded in the memory and then it is executed and the result is given back to the JavaWebServer and then it is transferred back to the client.

**JDBC**:

**JDBC connectivity**

The JDBC provides database-independent connectivity between the J2EE platform and a wide range of tabular data sources. JDBC technology allows an Application Component Provider to:

* Perform connection and authentication to a database server
* Manage transactions
* Move SQL statements to a database engine for preprocessing and execution
* Execute stored procedures
* Inspect and modify the results from Select statements

**APACHE TOMCAT**

#### **Getting Started**

Tomcat is a servlet container with a JSP environment. A servlet container is a runtime shell that manages and invokes servlets on behalf of users.

You can roughly partition servlet containers into:

**Stand-alone servlet containers**

These are an integral part of the web server. This is the case when using a Java-based web server, for example the servlet container that is part of the JavaWebServer. Stand-alone is the default mode used by Tomcat. Most web servers, however, are not Java-based, which leads us to the next two container types.

**In-process servlet containers**

The servlet container is a combination of a web server plugin and a Java container implementation. The web server plugin opens a JVM inside the web server's address space and lets the Java container run in it. If a certain request should execute a Servlet, the plugin takes control over the request and passes it (using JNI) to the Java container. An in-process container is suitable for multi-threaded single-process servers and provides good performance but is limited in scalability.

**Out-of-process servlet containers**

The servlet container is a combination of a web server plugin and a Java container implementation that runs in a JVM outside the web server. The web server plugin and the Java container JVM communicate using some IPC mechanism (usually TCP/IP sockets). If a certain request should execute a Servlet the plugin takes control over the request and passes it (using the IPCs) to the Java container. The response time of an out-of-process engine is not as good as in the in-process one but the out-of- process engine performs better in many measurable ways (scalability, stability, etc.).

### **Tomcat can be used as either a stand-alone container**

(Mainly for Development and debugging) or as an add-on to an existing web server (currently Apache, IIS and Netscape servers are supported). This means that whenever you are deploying Tomcat you will have to decide how to use it and, if you select options 2 or 3, you will also need to install a web server adapter.

**ORACLE:**

**Role Of Oracle In Database:**

ORACLE 8i is one of the many database services that plug into a client / server model. It works efficiently to manage resources, a database information, among the multiple clients requesting & sending.

**Structured Query Language (SQL)**

SQL is an inter-active language used to query the database and access data in database. SQL has the following features:

1. It is a unified language.
2. It is a common language for relational database
3. It is a non-procedural language.

**Introduction to Oracle:**

ORACLE is a comprehensive operating environment that packs the power of a mainframe system into user microcomputer. It provides a set of functional programs that user can use as tools to build structures and perform tasks. Because applications developed on oracle are completely portable to environment and then move it into a multi user platform. Users do not have to be an expert to appreciate ORACLE, but the better user understands the programmer, the more productivity and creativity you will use the tools it provides.

##### **The Oracle Environment:**

###### ORACLE is modular system that consists of the ORACLE database and several functional programs. ORACLE tools had four kinds of works:

* Database management
* Data access and manipulations
* Programming
* Connectivity

**Database Management Tools**

This usually (known as RDBMS by ORACLE) includes the core programs of Oracle’s database management system, the ORACLE database with its associated tables and views, which are stored in the Oracle’s data dictionary and a group of helpful activities. The data dictionary stores information related to every facet of database system. User names, user access rights, table storage information and auditing data for the disaster recovery are all stored in the data dictionary

# Data Access and Manipulations

All of Oracle’s data access and manipulation tools are firmly based on ANSI standard SQL. In Oracle, the tools that a user will use to access and manipulate data, has well as to design or use applications. Each provides separate point of entry and unique speech to the Oracle system.

SQL\*PLUS allows direct access to database with SQL FORMS offer a user-friendly way to create and use forms. SQL\*REPORT writer lets you to create formatted output. SQL\*MENU provides a way for you to integrate your application menus.

### **Oracle Supports Applications Deveopment**

SQL\*FORMS is an excellent user-friendly tool for quickly creating forms. User can start with extremely simple default forms or use the full screen painting function to create detailed screens foe accessing and updating multiple tables and for controlling and editing the data as is entered. In SQL FORMS, ORACLE provides unique control devices called TRIGGERS to influence user action on a field before, during and after date input. Those triggers can execute SQL commands, native SQL FORMS commands, or external procedural language sub routines from with a form.

These forms are an advanced fourth-generation tool that will adopt to your requirements extremely well. With SQL\*MENU you can link all of the forms. Programs and queries are easily maintained with the secured menu structures.

**Oracle Uses The Sql Command Set**

With SQL \* Menu you can like all forms, programs and queries in easily maintained secure structures.

Oracle provides a SQL command set that is close to the ANSI standard.ORACLE has added extensive report-formatting commands to extend the direct SQL language our capabilities and to delay the needs for alternative report formatting techniques. Statistical, Arithmetic string date/time functions are also included.

ORACLE has with limitation. The Single-user does not include all of the tools available on the hardware platforms. Also,ORACLE is relatively expensive and more complex than more single user, PC-basis database managers. ORACLE delivers a comprehensive package that allows for unlimited growth.

**Oracle Gives You Security and Control:**

ORACLE has several features that ensure the integrity of user database. If an interruption occurs in processing, a rollback can rest the database to a point before the disaster. If a restore is necessary, ORACLE has a roll forward command for creating a database to its most recent save point. Oracle provides users with several functions for serving data. Grant and Revoke commands limit access to information down to the row and column levels. Views are valuable future for limiting access to the primary tables in the database.

**Oracle Performs Completivity:**

ORACLE has been constantly improved to perform competitively on the largest database because RDBMS has been hampered by a repetition for slow access time. ORACLE had to prove itself continuously and so the unique feature of clustering techniques for storing data on the disk or another performance gained. Additional functions help control complex database installations. The active data dictionary, which automatically updates and logs modification to the database provide documentation data off loading form the modification process. Finally, ORACLE stores the DBMS kernel in extended memory , so more main memory is available for the application.

**Programming Tools**

One of the most important categories of tools available from oracle is its series of programming interface. This precompiled software provides a convenient and easy method of incorporating ORACLE SQL statement in high level programming language. Current oracle can interface with COBOL and ‘C’.

**HTML:**

Hypertext Markup Language (HTML), the languages of the World Wide Web (WWW), allows users to produces Web pages that include text, graphics and pointer to other Web pages (Hyperlinks).

HTML is not a programming language but it is an application of ISO Standard 8879, SGML (Standard Generalized Markup Language), but specialized to hypertext and adapted to the Web. The idea behind Hypertext is that instead of reading text in rigid linear structure, we can easily jump from one point to another point. We can navigate through the information based on our interest and preference. A markup language is simply a series of elements, each delimited with special characters, that define how text or other items enclosed within the elements should be displayed. Hyperlinks are underlined or emphasized works that load to other documents or some portions of the same document.

HTML can be used to display any type of document on the host computer, which can be geographically at a different location. It is a versatile language and can be used on any platform or desktop.

HTML provides tags (special codes) to make the document look attractive. HTML tags are not case-sensitive. Using graphics, fonts, different sizes, color, etc., can enhance the presentation of the document. Anything that is not a tag is part of the document itself.

**Basic HTML Tags :**

<A>……….</A> Creates hypertext links

<BODY>…</BODY> Contains all tags and text in the HTML document

<CENTER>...</CENTER> Creates text

<FONT>…</FONT> Formats text with a particular font

<FORM>...</FORM> Encloses a fill-out form

<FRAME>...</FRAME> Defines a particular frame in a set of frames

<H#>…</H#> Creates headings of different levels

<HEAD>...</HEAD> Contains tags that specify information about a document

<HR>...</HR> Creates a horizontal rule

<HTML>…</HTML> Contains all other HTML tags

<META>...</META> Provides meta-information about a document

<SCRIPT>…</SCRIPT> Contains client-side or server-side script

<TABLE>…</TABLE> Creates a table

<TD>…</TD> Indicates table data in a table

<TR>…</TR> Designates a table row

<TH>…</TH> Creates a heading in a table

**ADVANTAGES**

* A HTML document is small and hence easy to send over the net. It is small because it does not include formatted information.
* HTML is platform independent.
* HTML tags are not case-sensitive.

CHAPTER – 7

CONCLUSION

CONCLUSION

**Conclusion:**

The On Line Emart Solutions can be further developed into a separate, automated system with the following enhancements:

* A mail server can be implemented to send mails directly from the system to the inbox of the recipient. The code needed for the same is being implemented except the mail server.
* Help file can be included. The system, as of now, does not support any help facility for the users of the system. A help menu can be provided with a special function key and help command in the main page itself. Help can be either introduced as a separate window, a reference to a printed manual or as one or two line suggestion produced in a fixed screen location.

**SCOPE OF FUTURE ENHANCEMENT**

* Providing Mobile phone interaction for the current online project.

CHAPTER - 8

BIBLIOGRAPHY

BIBLIOGRAPHY

1. Roger S Pressman, **“Software Engineering – A Practitioner’s approach”** McGraw – Hill International Editions, Fifth Edition, 2001.
2. Henry F Korth, S. Sudharshan, **“Database System Concepts”** McGraw – Hill International Editions, Fourth Edition, 2002.
3. George Koch, Kevin Loney, **“Oracle – The Complete Reference”**, Tata McGraw Hill, Third Edition, 2001.
4. Herbert Schildt & Patrick Naughton, “**Java2 Complete Reference**”, Tmh 3/e, 1999.
5. James Jawroski, “**Mastering Java Script**”, Tmh 3/e, 2000.
6. George Mc Daniel, **“IBM Dictionary of Computing”**, Tata McGraw Hill, Tenth Edition, 1993.
7. Steven Holzner, **“HTML Black Book”,** TdreamTech Press,5th Edition, 2004.

**WEB SITES REFERRED:**

[**www.sun-java.com**](http://www.sun-java.com)

[**www.w3schools.org**](http://www.w3schools.org)

[**www.javacode.com**](http://www.javacode.com)

[**www.oracle.org**](http://www.oracle.org)

[**www.google.com**](http://www.google.com)