ONLINE WEBSHOPPING

A THESIS

SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENT FOR THE AWARD OF THE DEGREE OF

MASTER OF ENGINEERING
IN
COMPUTER SCIENCE AND ENGINEERING

TO

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY
(DEEMED UNIVERSITY)
PATIALA - 147 001

SUPERVISOR

Shri. Himanshu Aggarwal

SUBMITTED BY

M.HARI MOORTHY
REG. NO. ME-125/98(R)/1

DEPARTMENT OF COMPUTER SCIENCE
TECHNICAL TEACHERS' TRAINING INSTITUTE
SECTOR 26, CHANDIGARH-160019
2000

CERTIFICATE

Certified that this thesis report entitled "ONLINE WEBSHOPPING" submitted by Mr.M.Hari Moorthy in partial fulfillment of the requirement, for the award of Master of Engineering (Computer Science & Engineering) Degree of Thapar Institute of Engineering and Technology, Patiala is a record of student's own study carried under my supervision and guidance.

Thesis has not been submitted to any other University or institution for the award of any degree.

Supervisor

(Himanshu Aggarwal)
Lecturer
Department of Computer Science
T.T.T.I., Chandigarh

(V.P.Puri)
Prof.Information Management
& Coordination
Head
Department of Computer Science
T.T.T.I., Chandigarh

ACKOWLEDGEMENTS

The explosion of Internet and its applicability is all pervading. The project *Online Webshopping* project provided me an opportunity to update myself with Internet programming techniques and with emerging trends. The gain is indeed valuable.

But, it would have been impossible without the able guidance and encouragement of the project supervisor Mr.Himanshu Aggarwal, Lecturer, Dept. of Computer Science, TTTI, Chandigarh. I thank him profusely.

I express heartfelt thanks to Dr.Renu Vig, Asst.Professor, Dept. of Computer Science, TTTI, Chandigarh and M.E. Co-ordinator who was a source of inspiration and whose co-operation helped me complete the project.

I also thank Shri V.P.Buri, Professor, IMCO & H.O.D., Dept.of Computer Science, TTTI, Chandigarh for his kind dispensation in extending facilities without which the project could not have been completed in time.

I also gratefully acknowledge cooperation of faculty, other staff that went into making of this thesis.

M.HARI MOORTHY,

T.T.T.I., Chandigarh

ABSTRACT

Hitherto Internet is mainly used for sharing resources and data across the globe. It is now increasingly used for business and commerce applications. Against this backdrop, Online-shopping application is undertaken as project work.

Web design and e-commerce subtleties are enumerated and are followed by pictorial conceptualisation of the application. The context diagram, user interface and flow control (E-R diagram) supply clue as to how proceed for implementation.

Later, software approaches for client side and server side programming are discussed. It being net being application, HTML is used for the presentation of data to the user and JavaScript for client side validations of user responses.

Server side programming is implemented in JAVA using Servlets and RMI API's. The servlet API is very strong on server side and totally obleviates the use of CGI scripts.

The JDBC, Java API for database connectivity is also discussed briefly. The JDBC API provides classes and methods that help access and retrieval of data from underlying databases. The chapter on e-commerce is devoted to the discussion of standards, finer points of site design and about the site servers.

The project is implemented according to 3-tier paradigm namely the client, the intermediate server (middle tier) and the server.

ONLINE WEBSHOPPING

CONTENTS

Chapter 1

Online Shopping System

1.1.Introduction:

Every new technology introduced in this world has transformed every walk of human life. Now it is the turn of online business and e-commerce technologies unleashing sweeping changes to the way business is conducted. No wonder the advent of this will dictate and determine the course of human life in the new millennium. The proliferation of new infonomics already forcing the traditional economic theory to be rewritten.

The corporate world is under pressure from customers, partners, and shareholders to develop online capability. The growing web usage, well evolved IP standards, rapid application development tools, single market place – global reach are, among others, compelling reasons for organisations to go online. The online business has already reached its peak in the west and in India it is now catching up rapidly.

Against this backdrop, the Online Webshopping is taken up as project work and is implemented in Java, a much-favoured language for Internet programming. This shopping application can be adapted by any business enterprise with very little modifications.

1.2.System Characteristics:

Net based shopping application is radically different from traditional 'brick and mortar ' model of business. It is an integrated web based solution to the existing business practices.

It encompasses the amalgamation of the following entities.

*The customers, suppliers, vendors and service providers.

*The organisation's business practices and the technologies it employees

The characteristics of net shopping are

*The business is on 24 hrs a day and 365 days a year.

*The organization's products and services aredisplayed in the virtual market place(Complete transparency).

*The orders are going to pour from almost every part of the globe (Global market place).

*The customer wants a product according to his specifications known to him only yesterday (High expectations).

1.3.Prerequisite for application development:

Web design and online shopping application development is not one and the same though web design expertise is a prerequisite. Web design technologies and design experience come handy in the development of commerce application. Web design technologies comprise of software, protocols and vary from networks to fire walls, e- mail systems. Net shopping, on the other hand requires involves business strategies, available software and ability integrate them.

The application has the following stages

(i) Authoring & Designing:

HTML, DHTML, XML.

Graphics: Multimedia, audio, video and a animation.

(ii) Programming:

JavaScript, Java, ASP, application servers, Perl/CGI.

(iii) Backend:

Databases networks, security, platforms and protocols.

Chapter2

System Design

2.1. Problem definition:

The Online shopping application design involves interaction among different entities. To arrive at a design model, which can address atleast a minimum of issues, analysation of input, output and flow control is essential. Therefore system analysis is carried out with the help of diagrams. The context diagram, user interface and the flow control diagrams are drawn first. These form basis for deriving a plan for implementation. The input to the system come from user and output from the system is directed to the user, the banker and internally the stores, manufacturing unit . Apart from these, the system should carry and coordinate communication with various organs of a business enterprise.

2.2.Context Representation:

The foremost consideration in developing a system is how the system can be visualized so that it is near in its functionality to the real-life situation. The minimum functionality proposed for the online business application is outlined in the context diagram. Please refer to fig.2.1.The system reflects the important, crucial, activities of a small business enterprise..

The system is conceived such that it interacts with manufacturing unit, stores and the customer. System interaction with the customer is in terms of presenting product data, responding to customer queries, allowing the customer to place order in the case the customer is interested to etc.

The system interacts with manufacturing unit to find the availability of specified products, supply the same to the customer. Once the manufacturing unit okays the production of specified product (s) and it ensures that product consignment is kept in the store for later retrieval.

When the customer places the order, system takes careof payment and ensures the supply of consignment to the customer.

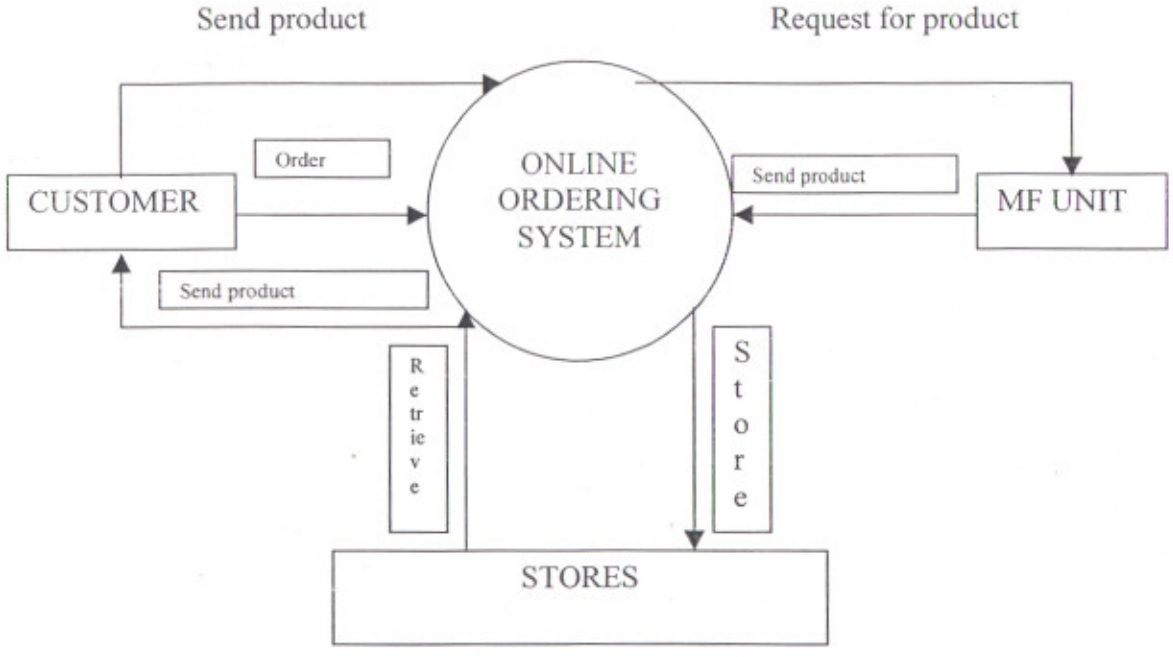
2.3.E-R Diagrams:

The fig 2.2.represents the relationships among the various entities of the proposed system. A visitor to the site is allowed to supply login data so that he can perform transactions any time thereafter. A customer so registered interacts, i.e. his relationship with other system units is reflected in the E-R diagram .In the system software development phase, these will be taken care of and implemented. Thus E-R diagram gives a total sum of activities the system fulfils.

2.4.Information Flow Control:

The flow of information in the proposed system is derived on the basis of E-R diagram is shown in fig.2.3. The customer and the business house are the two ultimate benefactors of the system. The flow of information from customer as monitored by the system is with regard to procuring product details, order placement. The flow of information from the enterprise consists of interaction with various sections and the customer.

Fig.2.1.Context diagram



MF UNIT--->MNUFACTURING Unit
The system is designed to communicate with three entities.

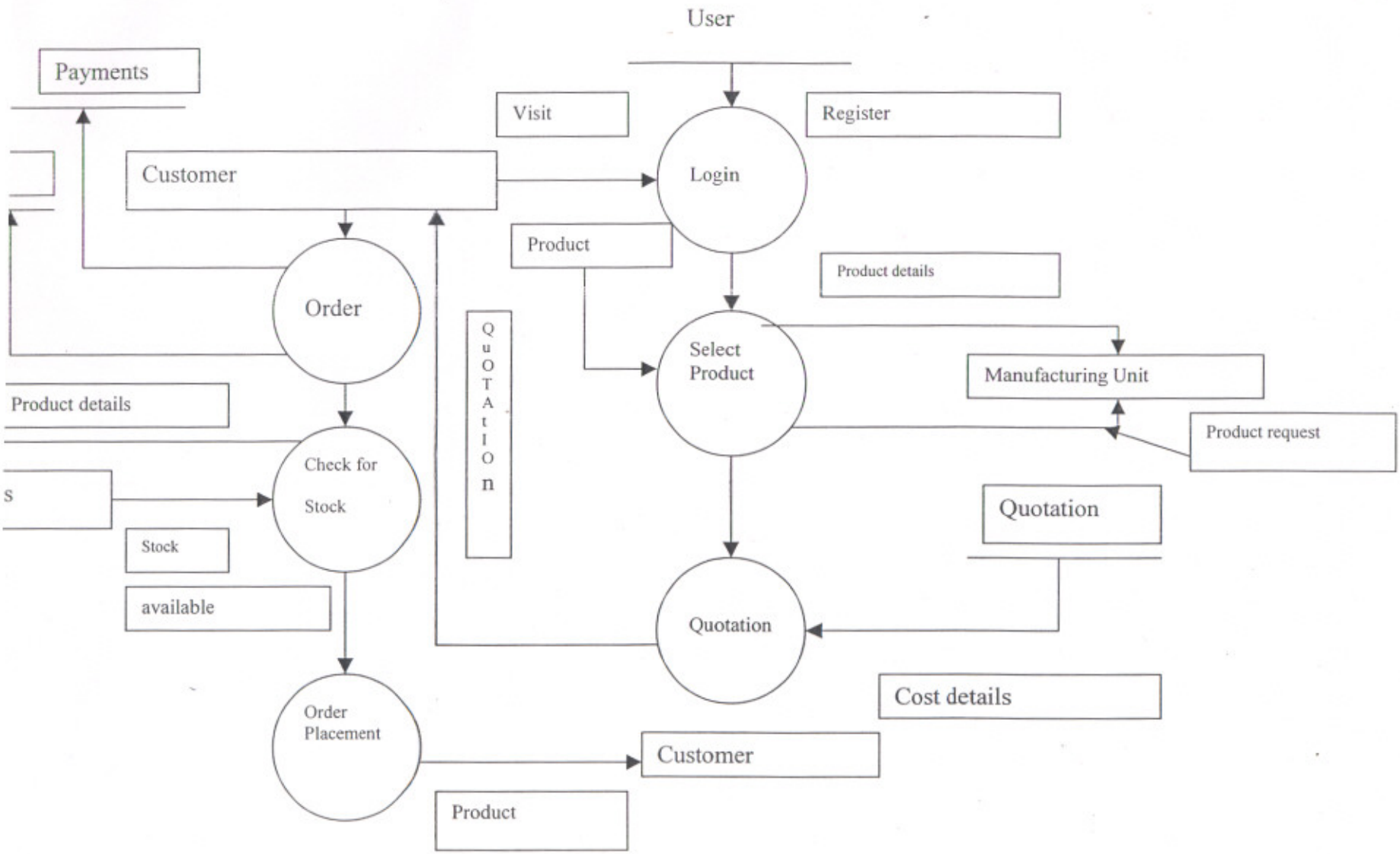
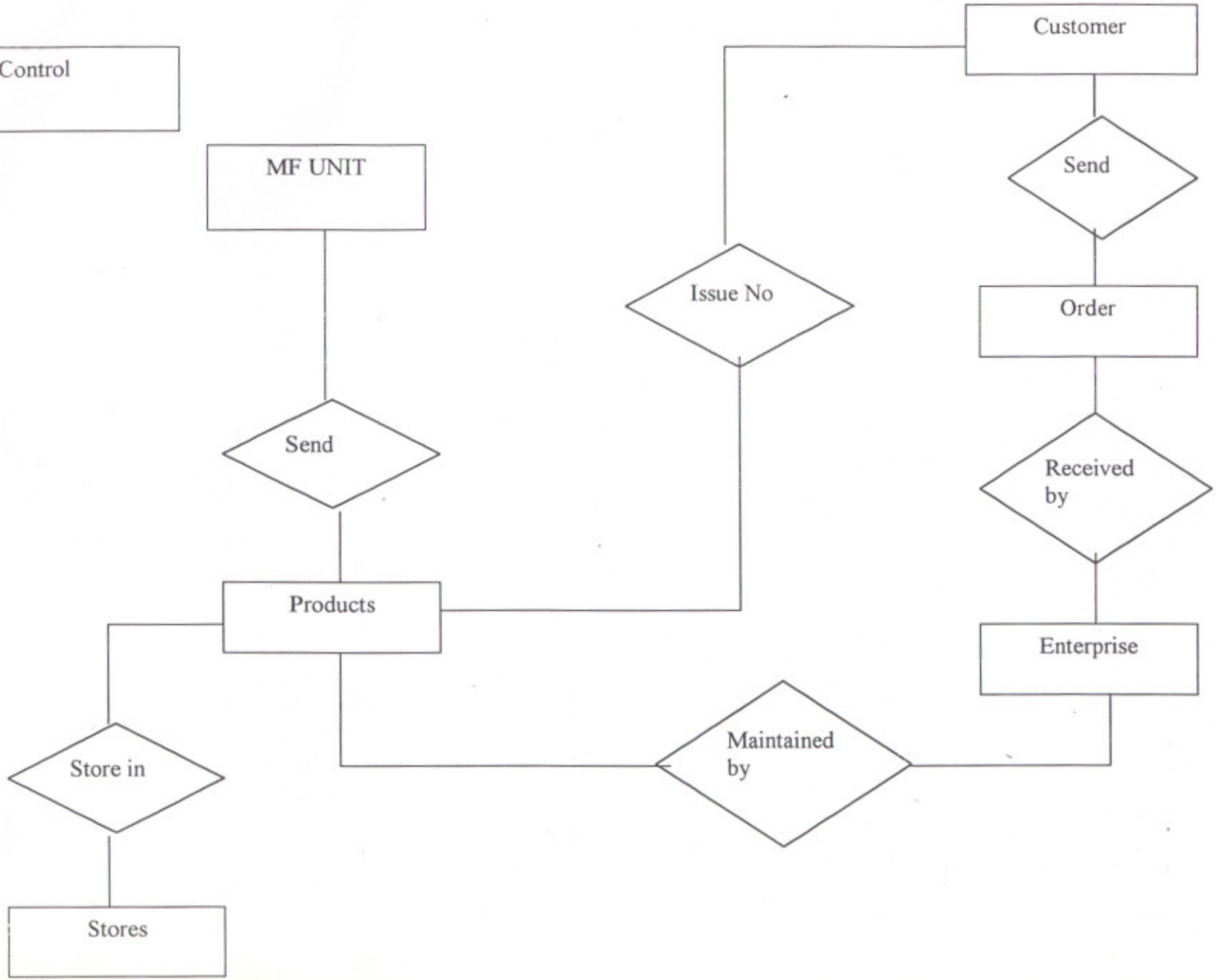


Fig2.1.E-R Diagram

2.3.Flow Control



Chapter 3

System Implementation

3.1.Introduction:

The project is net-based application. It involves client/server-programming considerations. For client/sever programming i.e. network programming Java has become choice of many. The system objects which signify certain functionality are developed using Java.The Java programs are called server programs. Then these objects are accessed using Java clients.(Clients developed in Java to invoke server objects).The data and information whatsoever comes from Java server to the Java client must be sent to the end user. The end user accesses it using the web browser. Therefore the information to be presented to the user must be HTML pages. These web pages are to be generated dynamically. The Java Servlet API is one, which does this. This part of the project implementation is discussed as Server side programming. The ultimate result to the user is web pages and so those tools that help present data as web pages are discussed as Client side tools.

3.2.Client side programming:

The user and the visitor to the site gains access to the site using web browser. The response to the client and data to be presented to the client must be as web pages so that browser can display. Therefore, HTML is used for client side programming.

The client, especially in a transaction oriented environment expected to supply information and answers to queries etc. There is a need to ensure that the user does not enter, inadvertently or deliberately, wrong data. This is called validation. Validation is carried out at client side rather than server side. This reduces network traffic and security risks apart from being faster. The client side validations implemented using Javascript.

3.3.Server side programming:

The server side programming is intended to address the issues like access and retrieval of Databases and performing necessary computations. This data again needs to be presented to the user as web pages as already discussed above. The server side programming is meant to say the programs that retrieve data from underlying databases and preset it to the end user. The Java API's help implement this and achieve the object are discussed as Server side programming tools. The databases that the system uses can be on the local machine, on a remote machine or distributed. The only requisite is using proper protocols and DSNs defined. Server side programming is, thus, implemented in Java.Java is the most favoured language for Internet programming. It is object-oriented, platform independent, robust and simple.

The Java servlet API, RMI API, and JDBC API come handy and make server side programming task much simpler. These Java API's come bundled with core Java API, JDK 2.0(Servlet API comes with javax package). These API's are subject matter of next chapters.

3.4.Databases:

Application needs vast data to store and process. Data requirements include storing of details of products, customer information, order placement and payment details. The DBMS used is Oracle databases.

3.5.System Specifications:

To run the Online shopping application, the required specifications are as under:

i). Web server that supports Servlets running on any platform,

ii). ODBC Driver installed,

iii). JDBC Drivers installed,

iv). JDBC-ODBC Bridge installed,

v). Pentium system (233 MHz minimum)

6.4 GB HDD, 128 MB RAM,

vi). Connectivity.

Client:

i). Workstation windows compatibility,

ii). 8 MB RAM,

iii). Java-enabled browser.

3.6.User Interface:

Whenever the user accesses the site and intents to interact with, homepage is displayed. On clicking the 'Next" option on Homepage, a window with two frames is displayed. The bottom window contains buttons with proper labels. The user can get product details, the list of customers registered or form for order placement on choosing appropriate buttons. The user will not be allowed to place order if he is not a registered customer and a registered customer failing to supply the correct data. Since user is provided with a GUI interface, navigation is quite easy.

Chapter 4

CLIENT SIDE TOOLS

4.1.Introduction:

Client side tools as mentioned in the pervious chapter are meant to mean the tools, which help present the content of data accessed and made available to the user. The Java server programs generate dynamically with help of these tools. Each of these tools is explained briefly.

4.2. HTML:

HTML stands for HyperText MarkUp Language and is used to create web pages. It is a very simple computer language. It is used to "mark-up" a whole load of text and therefore make it look better. It allows for images or sound to be used in web pages, for text to be presented and formatted effectively and more! Although originally planned to do nothing but add a structure to a document.

Word Processing changes can also be made to text in an HTML document except that they are slightly more complicated to enforce. There are lots of possibilities of what can be done with HTML.

HTML is about marking up text, this is done using "HTML tags". It is not necessary to have text, some pages are made up of nothing but tags. A tag starts with the '<' sign and ends with a '>' sign. Between these we place text which identifies which tag we are using and customises that tag for our needs. An example of a tag would be <html>, which marks the beginning of the html document. The html tag is a container. A container is made up of a start tag and an end tag. And whatever appears in between the two tags, which can be effected, is effected in whatever way the tag is designed to effect it.

Some of these Tags Are:

<HEAD>...</HEAD>

<TITLE>...</TITLE>

<META>..</META>

<APPLET<...</APPLET>

<!- -this is the text that is hidden- ->

<BODY>...</BODY>

<PRE>...</PRE>

<H3>...</H3> ETC..,

4.3.1.JavaScript:

JavaScript is a compact, object-based scripting language for developing client and server Internet applications.

JavaScript is case sensitive. Netscape Navigator Version 2.0 onwards interprets JavaScript statements embedded directly in an HTML page. In a client application for Navigator, JavaScript statements embedded in an HTML page can recognise and respond to user events such as mouse clicks, form input, and page navigation.

For example, written a JavaScript function to verify that users enter valid information into a form requesting a telephone number or zip code. Without any network transmission, an HTML page with embedded JavaScript can interpret the entered text and alert the user with a message dialog if the input is invalid. Or you can use JavaScript to perform an action (such as play an audio file, execute an applet, or communicate with a plug-in) in response to the user opening or exiting a page.

4.3.2.Comparision between JavaScript and Java:

The JavaScript language resembles Java, but without Java's static typing and strong type checking. JavaScript supports most of Java's expression syntax and basic control flow constructs. In contrast to Java's compile-time system of classes built by declarations, JavaScript supports a run-time system based on a small number of data types representing numeric, Boolean, and string values. JavaScript has a simple instance-based object model that still provides significant capabilities. JavaScript also supports functions, again without any special declarative requirements. Functions can be properties of objects, executing as loosely typed methods. JavaScript complements Java by exposing useful properties of Java applets to script authors.

JavaScript statements can get and set exposed properties to query the state or alter the performance of an applet or plug-in. Java is an extension language designed, in particular, for fast execution and type safety. Type safety is reflected by being unable to cast a Java into an object reference or to get at private memory by corrupting Java bytecodes.

Java programs consist exclusively of classes and their methods. Java's requirements for declaring classes, writing methods, and ensuring type safety make programming more complex than JavaScript authoring. Java's inheritance and strong typing also tend to require tightly coupled object hierarchies.

In contrast, JavaScript descends in spirit from a line of smaller, dynamically typed languages like

HyperTalk and dBase. These scripting languages offer programming tools to a much wider audience because of their easier syntax, specialised built-in functionality, and minimal requirements for object creation.

4.3.3.The following table compares and contrasts JavaScript and Java.

JavaScript	*Java*
Interpreted (not compiled) by client.	Compiled on server before execution on client
Object-based. Code uses built-in, Extensible objects, but not of object classes or inheritance and embedded in, HTML	Object-oriented. Applets consist of objects classes with inheritance, inheritance Code integrated with Applets distinct from HTML (accessed from HTML pages).
Variable data types not declared (loose typing).	Variable data types must be declared (strong typing).

Dynamic binding. Object references checked at run-time	Static binding. Object references must exist at compile- time.
Secure. Cannot write to hard disk.	Secure. Can write to hard disk.

4.3.4.Using JavaScript in HTML:

JavaScript can be embedded in an HTML document in two ways:

As statements and functions using the SCRIPT tag.

As event handlers using HTML tags.

The SCRIPT tag

A script embedded in HTML with the SCRIPT tag uses the format:

```
<SCRIPT>
    JavaScript statements...
</SCRIPT>
```

The optional LANGUAGE attribute specifies the scripting language as follows:

```
<SCRIPT LANGUAGE="JavaScript">
  JavaScript statements...
</SCRIPT>
```

The HMTL tag, <SCRIPT>, and its closing counterpart, </SCRIPT> can enclose any number of JavaScript statements in a document.

4.3.5.JavaScript Event Handling:

Event	occurs when...	Event Handler
Blur	User removes input focus from form element	onBlur
click	User clicks on form element or link	onClick

change	User changes value of text, textarea, or select element	onChange
focus	User gives form element input focus	onFocus
load	User loads the page in the Navigator	onLoad
mouseover	User moves mouse pointer over a link or anchor	onMouseOver
select	User selects form element's input field	onSelect
submit	User submits a form	onSubmit
unload	User exits the page	onUnload

The JavaScript Language contains the following built-in objects and functions:

1. String object
2. Math object
3. Date object
4. Built-in functions ETC.,

4.4.CGI:

"CGI" stands for "Common Gateway Interface". CGI is the method by which a web server can obtain data from (or send data to) databases, documents, and other programs, and presents that data to viewers via the web. More simply, CGI is programming for the web. A CGI can be written in any programming language, but PERL is the most popular.

4.5.Web Development Environment:

Editing: Access local and remote files, insert and convert files, set bookmarks, easy tag selection, automatically convert special characters.

Page Design: Visually layout page elements for quick edits and rapid prototyping. Convert content, such as lists and tables, from Microsoft Word and Excel into equivalent HTML formatting.

CodeSweepers: Enforce code formatting with customisable rules.

Browsers: Preview pages in multiple browsers, automatic detection of installed browsers

Proofing and testing tools: Maintain site content with search and replace, spell checking, code validation, and link verification.

Enhanced Projects: Create, deploy, and manage local and remote Web sites.

Tag Inspector: Edit code in a customisable property sheet interface.

Tag Tree : View and navigate the current document's tag hierarchy.

Site View : View, edit, and test links on the current page and in linked pages.

Style Editor: Easily create, edit, preview, and link Cascading Style Sheets.

Code templates: Quickly inserts common text blocks and expand abbreviations.

Snippets: Save code blocks and content for reuse.

Templates and Wizards: Quickly creates basic pages, tables, frames, JavaScript and DHTML elements, and synchronised Real Audio content.

Image preview: View Web images and their properties.

Chapter 5

Server Side Tools

5.1.Introduction:

The tools with which the server side programming is carried out are Java RMI, Servlets and JDBC. RMI is described in this chapter and Servlets and JDBC in separate chapters.

5.2:Remote Method Invocation [RMI]

5.2.1.Distributed systems:

Distributed systems require that computations running in different address spaces, potentially on different hosts, be able to communicate. For a basic communication mechanism, the JavaTM language supports sockets, which are flexible and sufficient for general communication. However, sockets require the client and server to engage in applications-level protocols to encode and decode messages for exchange, and the design of such protocols is cumbersome and can be error-prone. An alternative to sockets is Remote Procedure Call (RPC), which abstracts the communication interface to the level of a procedure call. Instead of working directly with sockets, the programmer has the illusion of calling a local procedure, when in fact the arguments of the call are packaged up and shipped off to the remote target of the call. RPC, however, does not translate well into distributed object systems, where communication between program-level objects residing in different address spaces is needed. In order to match the semantics of object invocation, distributed object systems require remote method invocation or RMI. In such systems, a local surrogate (stub) object manages the invocation on a remote object.

The goals for supporting distributed objects in the Java language are:

1. Support seamless remote invocation on objects in different virtual machines.

2. Support callbacks from servers to applets.

3. Integrate the distributed object model into the Java language in a natural way while retaining most of the Java language's object semantics.

4. Make differences between the distributed object model and local Java object model apparent.

5. Make writing reliable distributed applications as simple as possible.

6. Preserve the type-safety provided by the Java runtime environment.

7. Various reference semantics for remote objects; for example live (nonpersistent)references, persistent references, and lazy activation.

8. The safe Java environment provided by security managers and class loaders.

5.2.2.Distributed object applications:

RMI: Remote method invocation (RMI) is the action of invoking a method of a remote interface on a remote object. Most importantly, a method invocation on a remote object has the same syntax as a method invocation on a local object.

RMI applications are often comprised of two separate programs: a server and a client. Typical server application creates a number of remote objects, makes references to those remote objects accessible, and waits for clients to invoke methods on those remote objects. A typical client applications gets a remote reference to one or more remote objects in the server and then invokes methods on them. RMI provides the mechanism by which the server and the client communicate and pass information back and forth. Such an application is sometimes referred to as a distributed object application.

5.2.3.Distributed object applications need to:

Local Remoted Objects: Applications can use one of two mechanisms to obtain references to remote objects. An application can register its remote objects with RMI's simple naming facility, the rmiregistry, or the application can pass and return remote object references as part of its normal operation.

5.2.4.Communication with remote objects:

Details of communication between remote objects are handled by RMI; to the programmer, remote communication looks like a standard Java method invocation. Load class bytecodes for objects that are passed as parameters or return values. Because RMI allows a caller to pass pure Java objects to remote objects, RMI provides the necessary mechanisms for loading an object's code as well as transmitting its data.

5.2.5.Overview of RMI interfaces and classes:

The interfaces and classes that are responsible for specifying the remote behaviour of the RMI system are defined in the java.rmi package hierarchy.

5.2.6.The java.rmi.Remote Interface:

In RMI, a remote interface is an interface that declares a set of methods that may be invoked from a remote Java virtual machine.

A remote interface must satisfy the following requirements:

A remote interface must at least extend, either directly or indirectly, the interface java.rmi.Remote.

Each method declaration in a remote interface must satisfy the requirements of a remote method declaration as follows:

A remote method declaration must include the exception java.rmi.RemoteException (or one of its superclasses such as java.io.IOException or java.lang.Exception) in its throws clause, in addition to any application-specific exceptions (note that application specific exceptions do not have to extend java.rmi.RemoteException).

In a remote method declaration, a remote object declared as a parameter or return value (either declared directly in the parameter list or embedded within a non-remote object in a parameter) must be declared as the remote interface, not the implementation class of that interface.

5.2.7.The Remote Object Class and its Subclasses:

Java.rmi.server.RemoteObject and its subclasses, java.rmi.server.RemoteServer and java.rmi.server.UnicastRemoteObject and java.rmi.activation.Activatable provide RMI server functions.

5.2.8.Implementing a Remote Interface:

The general rules for a class that implements a remote interface are as follows:

The class usually extends java.rmi.server.UnicastRemoteObject, thereby inheriting the remote behaviour provided by the classes Java.rmi.server.RemoteObject and java.rmi.server.RemoteServer. The class can implement any number of remote interfaces. The class can extend another remote implementation class. The class can define methods that do not appear in the remote interface, but those methods can only be used locally and are not available remotely.

Chapter 6

Servlets

6.1.Introduction:

Java servlets are classes that can be loaded dynamically to expand the functionality of a server. Used within web server to take the place CGI scripts. Servlets are safe and portable. Unlike applets, they do not require support for Java in the browser. CGI use multiple processes to handle separate programs and/or requests but servlets are handled by separate threads within the web server. Servlets are efficient and scalable. Servlets run in the server, can interact closely and do things not possible with CGI scripts. Servlets are portable, across operating systems and across web servers.

6.2.Overview of Servlets:

Servlets are modules that extend request/response-oriented servers, such as Java-enabled web servers. For example, a servlet might be responsible for taking data in an HTML order-entry form and applying the business logic used to update a company's order database.

Servlets are to servers what applets are to browsers. Unlike applets, however, servlets have no graphical user interface. Servlets can be embedded in many different servers because the servlet API, which help write servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers.

6.3. Servlets instead of CGI Scripts:

Servlets are an effective replacement for CGI scripts. They provide a way to generate dynamic documents that is both easier to write and faster to run. Servlets also address the problem of doing server-side programming with platform-specific APIs: they are developed with the Java Servlet API, a standard Java extension. So use servlets to handle HTTP client requests. For example, have servlets process data

POSTed over HTTPS using an HTML form, including purchase order or credit card data. A servlet like this could be part of an order-entry and processing system, working with product and inventory databases, and perhaps an on-line payment system.

A servlet can handle multiple requests concurrently, and can synchronise requests. This allows servlets to support systems such as on-line conferencing.

6.4.Other Uses:

Forwarding requests: Servlets can forward requests to other servers and servlets. Thus servlets can be used to balance load among several servers that mirror the same content, and to partition a single logical service over several servers, according to task type or organisational boundaries.

6.5.The Servlet Package:

The javax.servlet package provides interfaces and classes for writing servlets. The architecture of the package is described below.

The Servlet Interface:

The central abstraction in the Servlet API is the Servlet interface. All servlets implement this interface, either directly or, more commonly, by extending a class that implements it such as HttpServlet.

The Servlet interface declares, but does not implement, methods that manage the servlet and its communications with clients. Servlet writers provide some or all of these methods when developing a servlet.

6.6.Client Interaction:

ServletRequest, which encapsulates the communication from the client to the server. A ServletResponse, which encapsulates the communication from the servlet back to the client. ServletRequest and ServletResponse are interfaces defined by the javax.servlet package.

When a servlet accepts a call from a client, it receives two objects:

- The ServletRequest Interface:
- The ServletRequest interface allows the servlet access to:

Information such as the names of the parameters passed in by the client, the protocol (scheme) being used by the client, and the names of the remote host that made the request and the server that received it.

The input stream, ServletInputStream. Servlets use the input stream to get data from clients that use application protocols such as the HTTP POST and GET methods.

Interfaces that extend ServletRequest interface allow the servlet to retrieve more protocol-specific data. For example, the HttpServletRequest interface contains methods for accessing HTTP-specific header information.

The ServletResponse Interface

The ServletResponse interface gives the servlet methods for replying to the client.

It:

- Allows the servlet to set the content length and MIME type of the reply.
- Provides an output stream, ServletOutputStream, and a Writer through which the servlet can send the reply data.

Interfaces that extend the ServletResponse interface give the servlet more protocol-specific capabilities. For example, the HttpServletResponse interface contains methods that allow the servlet to manipulate HTTP-specific header information.

6.7.Additional Capabilities of HTTP Servlets

The classes and interfaces described above make up a basic Servlet. HTTP servlets have some additional objects that provide session-tracking capabilities. The servlet writer can use these APIs to maintain state between the servlet and the client that persists across multiple connections during some time period. HTTP servlets also have objects that provide cookies. The servlet writer uses the cookie API to save data with the client and to retrieve this data.

6.8.A Simple Servlet:

The following class completely defines servlet:

```
public class SimpleServlet extends HttpServlet
  {
    /**
     * Handle the HTTP GET method by building a simple web page.
     */
    public void doGet (HttpServletRequest request,
                       HttpServletResponse response)
    throws ServletException, IOException
    {
     PrintWriter out;
     String title = "Simple Servlet Output";

     // set content type and other response header fields first
       response.setContentType("text/html");

    // then write the data of the response
       out = response.getWriter();

       out.println("<HTML><HEAD><TITLE>");
       out.println(title);
       out.println("</TITLE></HEAD><BODY>");
       out.println("<H1>" + title + "</H1>");
       out.println("<P>This is output from SimpleServlet.");


       out.println("</BODY></HTML>");
       out.close();
    }
```

```
}
```

The nitty-gritty of the above example:

SimpleServlet extends the HttpServlet class, which implements the Servlet interface.

SimpleServlet overrides the doGet method in the HttpServlet class. The doGet method is called when a client makes a GET request (the default HTTP request method), and results in the simple HTML page being returned to the client.

Within the doGet method,

The user's request is represented by an HttpServletRequest object.

The response to the user is represented by an HttpServletResponse object.

Because text data is returned to the client, the reply is sent using the Writer object obtained from the HttpServletResponse object.

Chapter7

The Java Database Connectivity

7.1.Introduction:

To communicate with data storehouse pertaining to any application an appropriate API is needed. The JDBC is part of Java Enterprise API. The JDBC is a set of relational database objects and methods that provide access to SQL data sources. JDBC is based on SQL call level interfaces. The JDBC classes are in the java.sql.package.

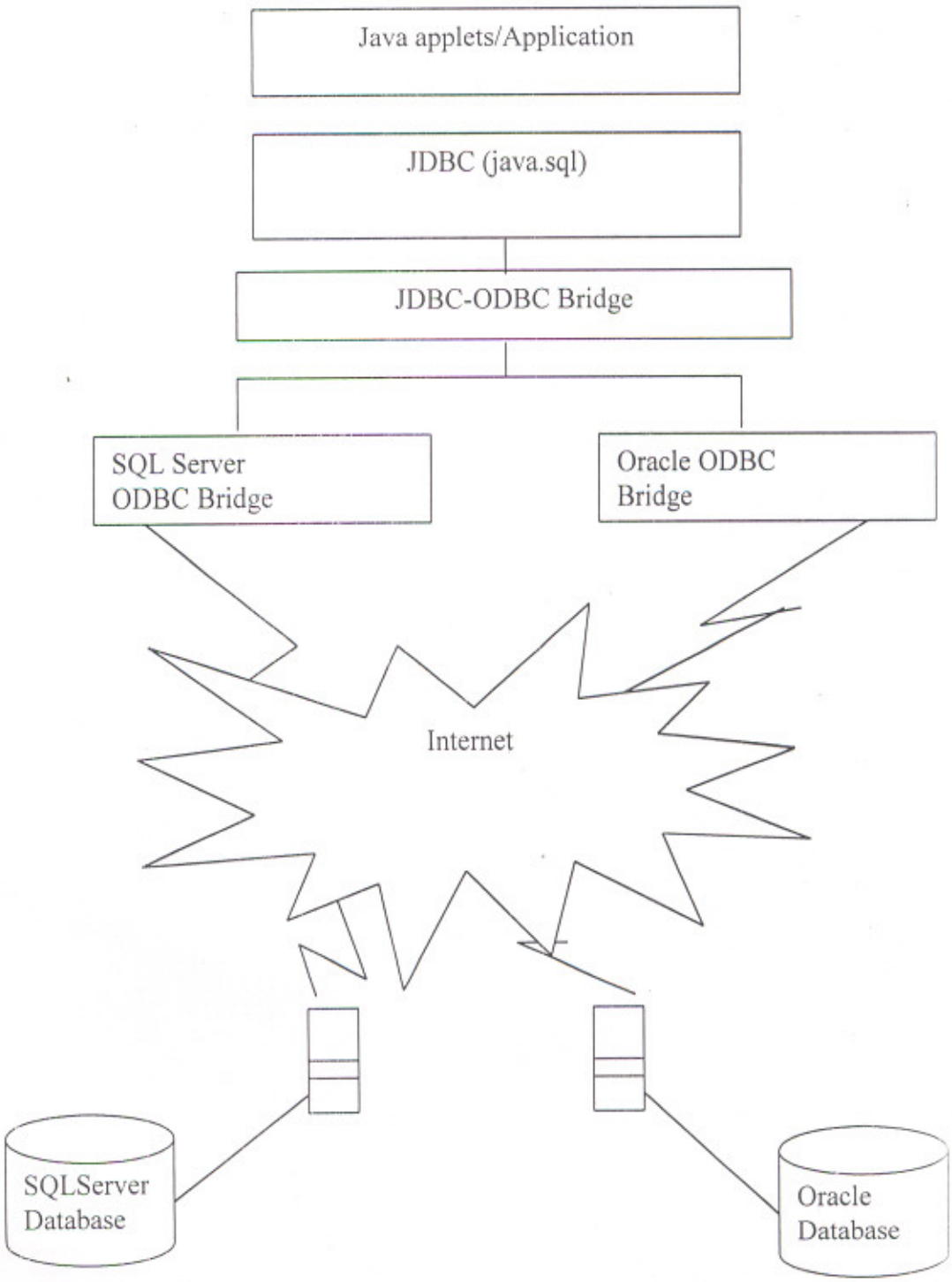
The JDBC database is designed using three-tier model. In three-tier paradigm, the middle server contains the required business logic thereby isolating the client from server. This approach has many advantages like; the client will not be affected even if data sources change.

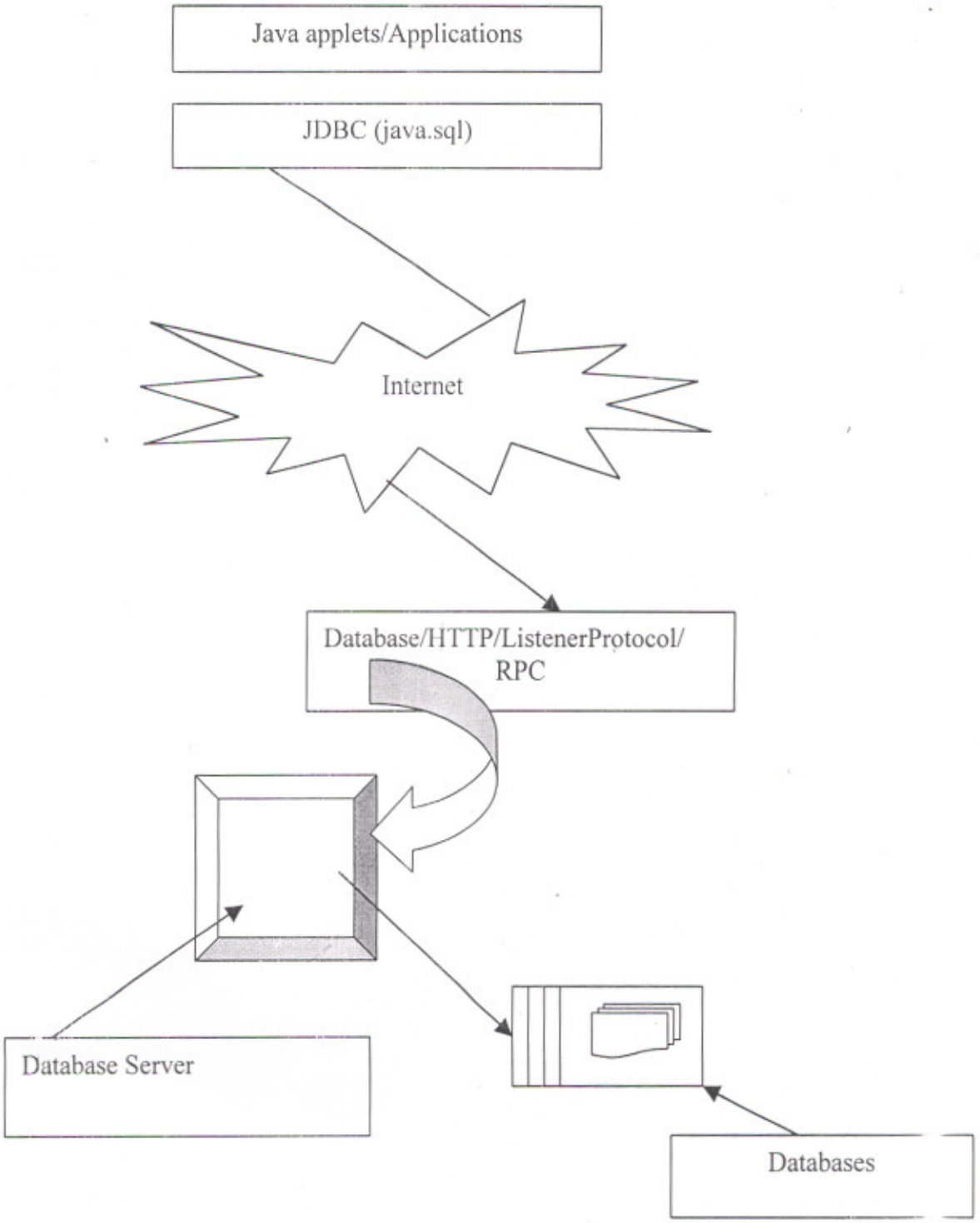
7.2.JDBC Overview:

JDBC defines a set of API objets and methods to interact with underlying databases Java program first opens a connection to a database, makes a statement object and retrieves the results as well as metadata. To reduce latency during execution, it is better to have JDBC classes in the client (like applets).

7.3. The JDBC Communication layer alternatives:

Scheme (i).




Java applets/Applications
JDBC (java.sql)
Internet
Database/HTTP/ListenerProtocol/
RPC
Database Server
Databases

Scheme (ii).

A program using JDBC needs a driver for the the particular datasource with which it wants to interface. Can be a native module or it can be a Java program that talks to a server in the network by using some RPC or Listener-talker protocol.Both schemes are shown above.

It is conceivable that an application will deal with more than one data source, a heterogeneous data sources.The DriverManager manages the drivers and provides a list of currently loaded drivers to the application programs.

7.4..JDBC-ODBC Bridge:

As depicted in scheme (ii), a Java program using JDBC must implement the native driver of that data source but facilitate accessing multiple data sources, a common interface is required.Sun following design principle of ODBC, developed JDBC-ODBC Bridge. JDBC-ODBC Bridge converts JDBC calls into ODBC compliant native calls.(ODBC is a C interface which implements many native drivers.) Please refer to Fig.7.4.i.

7.5.JDBC classes – Overview:

The topmost class in the hierachy of java.sql package is DriverManager.The DriverManager keeps the driver information, state information etc.When each driver is loaded, it registers with the DriveverManager.The DriverManager when required to open a connection,selects the driver depending on the JDBC URL.

The java.sql.Drivere class is reffered to for information such as PropertyInfo, version number etc.This class could be loadedd many times during program execution.

7.6.JDBC Apllication:

To handle data from a database, a Java program calls the getConnection() method which returns Connection object.Then creates a Statement object and prepares SQL statement. A SQL statement can be a Statement object, PreparedStatement object(compiled statement) or can be a CallableStatement

object(stored procedure).When method executeQuery() is executed, a ResultSet object is returned. SQL statements update, delete will not return a ResultSeet. For such statements executeUpdate() method is used.

The ResultSet contains rows of data that are parsed using the next() method.

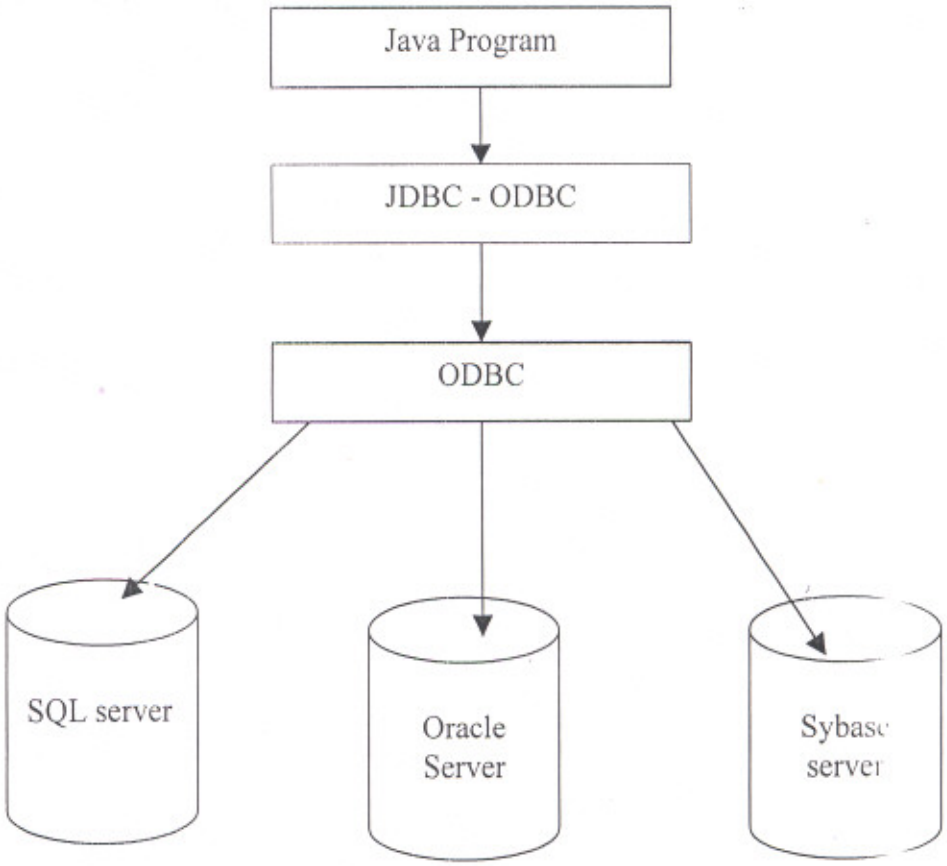


Fig.7.4.i. JDBC architecture

Chapter 8

E-Commerce - An Overview

8.1. E-commerce Standards:

Like Internet standards and in addition to ,, e-commerce employs several of its own standards, most of which apply to business-to-business transactions.

Electronic Data Interchange (EDI):

Created by the government in the early 1970s and now used by 95 % companies, EDI is a common document structure designed to let large organizations transmit information over private networks. EDI is now finding a role on corporate Web sites as well.

Open Buying on the Internet (OBI):

This standard, created by the Internet Purchasing Roundtable, is supposed to ensure that all the different e-commerce systems can talk to one another. OBI, which was released by the OBI Consortium, is backed by leading technology companies such as Actra, InteliSys, Microsoft, Open Market, and Oracle.

The Open Trading Protocol (OTP):

OTP is intended to standardize a variety of payment-related activities, including purchase agreements, receipts for purchases, and payments. It was created as a competing standard to OBI by a group of companies, including AT&T, CyberCash, Hitachi, IBM, Oracle, Sun Microsystems, and British Telecom.

The Open Profiling Standard (OPS):

A standard backed by Microsoft and Firefly, OPS lets users create a personal profile of preferences and interests that they want to share with merchants. The idea behind it is to help consumers protect their privacy without banning online collection of marketing information.

Secure Sockets Layer (SSL):

This protocol is designed to create a secure connection to the server. SSL uses public key encryption, one of the strongest encryption methods around, to protect data as it travels over the Internet. SSL was created by Netscape but has now been published in the public domain.

Secure Electronic Transactions (SET):

SET encodes the credit card numbers stored on merchants' servers. This standard, created by Visa and MasterCard, enjoys wide support in the banking community. The first SET-enabled commerce is already being tested in Asia.

Trust:

This partnership of companies seeks to build public trust in e-commerce by putting a *Good Housekeeping*-style seal of approval on sites that don't violate consumer privacy.

8.2.E-commerce Software-General perspective:

Alongside the surge in e-commerce, the slew of catalogue building software options continues to burgeon. E-commerce software are also burgeoning making the choice difficult. Usually, mid-level e-commerce solutions might not suit the small enterprises. At a bare minimum, what is required is maintenance of own Web server to run these programs. Encanto's E.go is an example of an affordable choice. Essentially it's a Java-based Web server with e-commerce functionality that can work over a dial-up connection. E.go's only requisition for credit card authorization is that the marketer has own domain name and digital certificate.

Popular mid-level e-commerce solutions are IBM Net.Commerce, iCat, and Microsoft's Siteserver Commerce. IBM's package lets creation of catalogues from three prototypes, yet its drawback is it only supports Oracle databases. For the use of Windows NT, one might want to consider Microsoft's BackOffice compatible e-commerce software.

Corporate giants, on the other hand, may need to step-up their software to high-end powerhouses like Netscape EDI based CommerceXpert. Another large-scale option is Broad Vision's turnkey One to One Commerce solution. The ' iCat' has been around for a while and it's practical for small to medium sized businesses. The iCat runs on Personal Web Server on Windows NT 4.0. Solaris, IRIX, and HP-UX.

8.3.Design Customisation:

No doubt e-commerce is catching up, by and large, but still not became profitable. Therefore, in addition picking up best development packages in terms of commercial viability and functionality, the developer and the enterprise should see that a visitor to the becomes a buyer. Here are important things to remember while designing an online store.

8.3.1.Make Buyers Feel Comfortable:

Many Internet users are uncomfortable with the idea of purchasing goods online. Removal of customer apprehensions lead one step closer to winning a customer.

8.3.2.Good Street Cred:

To succeed, site needs to be perceived as reputable and credible. Creation of a presence that says a lot about site's seriousness and commitment go a long way in building a successful online business. If users see a site where the details are in order, it creates the impression the business behind it is run by a reputable merchant This means refining the user experience, focusing on the tasks and information-gathering activities that users go through. For example, if a site's strong suit is its vast catalogue and product availability, its benefits must represented prominently in the design. Avoidance of distractions that make the site look cheap, such as cheesy graphics, bad clip art, flaming logos, and blinking text. The site can be tasteful without being fancy or cluttered.

8.3.3.The Difference between *Buy* and *Bye:*

Consistent editorial tone always complements the purpose of the site, and keeps the content fresh. The text unifies and creates consistency throughout the site. It must be ensured that it is proof-read and content is polished, especially on the home page and on other frequently visited pages. The site littered with typos, communicates negative messages.

Let the site never get stale. Keeping dated information, such as a News section that hasn't been updated in last few years and things like that, may even make the user to thing that the company is out of business.

8.3.4.Lay Down the Law:

A reputable e-commerce site publishes clearly defined policies; the most important of these are about security/fraud protection, returns, and privacy. These policies let users know the site has a consistent and documented set of procedures. Also, they should be concise and clear; these aren't legal briefs, it is just educating consumers. Even if people never read the policies, they often find comfort in simply knowing they exist.

8.3.5.Transparency:

The personalities, the identities of the persons running the business must be transparent to the customers, letting them know the real people working at the business is always rewarding. The site be incorporated with links that provide easy-to-find, contact and customer support information, perhaps as often as every page. Phone numbers and email addresses are helpful. Even though much snail mail is not expected, a postal address lets users know a physical location exists, which is valuable in establishing a reputation as a legitimate company. When the company produces correspondence, either through customer service or transaction completion notifications address the user in a professional yet personal tone. All activities and transactions must be carried out in a timely fashion; without making them feel like just another faceless order number.

8.3.6.Keep Users Interest Alive:

First-time buyer promotions and incentives will make users think they've found deals they can't pass up. When people think they're about to score, they'll lower their guards and become eager to complete the transaction. And, since they think they're getting such a good deal, their perceived risk has been lowered. Common promotions to attract new customers include discounts when a purchase reaches a specific amount (for example, $20 off orders of $50 or more), free shipping, free returns, or first-time discounts (15 percent off your first order).

Web commerce isn't so much about technology as it is about selling. That may sound obvious, but while vendors are trying to focus on buying their e-commerce solutions to solve all business man's your problems, most online marketers would be better off focusing r energy on the design of the selling environment. Products are important, but there's **no substitute for carefully planned site navigation, a fully thought-out ordering and fulfilment process, procedures to ensure customer satisfaction, and fanatical quality assurance**.

A marketer trying to sell online, got to consider a lot. How does traditional merchandising work on the Web? Whether to use customer credit cards or any other mode of payment. It does not matter if the purchasing form is called a shopping *basket* or a shopping *cart*.? (Some online merchants think it does.)

8.4.Site Servers – An Overview:

8.1.Introduction:

To day a number of tools and what are called site server and commerce server are available in the market. These tools make earlier online shopping development less tedious and very easy to manage.

8.4.2.Evolution of Site Servers:

Site server technology origins lie in HTML static content augmented with CGI. This model served well for a while; however, development of a new store was a long process of modifying both the HTML content and all the CGI programs.

The purchasing process was tailored specially for the needs of an individual online store. Business rules and storefronts were hard-coded and designed, more or

less, from scratch. A number of commercial sites exist on the Net today that employ these earlier technologies. Because of the large effort required to build a minimal system, most of these sites do not have an easy-to-use set of administrative functionality built into the base store.

Working with the online stores required knowledge of HTML and CGI programming. In contrast to this approach today's site servers provide an ideal world where one only need to update assets such as products (product description) and images to modify or build a store.

8.4.3.Site servers functions:

The site servers and commerce servers, in general perform the following functions. However, a note must be made that all the servers available in the market may not offer some of these or some of the them can have many more features. Hence the following is only suggestive.

Stores based on these early technologies were typically stand-alone, meaning they had few (or no) interfaces to other systems and applications. Many a site servers contain commerce server component. Most of these servers expected to work coherently with other existing systems.

1. Permit remote build of stores,
2. Handle heavy processing loads efficiently, extends by means of published application program interfaces (APIs), integrate technologies and products, and presents a flexible architecture that supports diverse system configurations. This means that companies can readily translate their existing applications and information-technology infrastructure into Internet-based marketing. It also means that any third-party service provider can easily build a component that plugs in to the system.
3. Provide a means of collecting and storing shopper data, managing membership accounts, and creating customized site reports. A company can quickly create targeted discount promotions based on detailed user tracking and analyze the promotion results for future planning. This new technology improves a merchant's ability to offer customers attractive

sales opportunities, improve product introduction strategy, and provide a customized environment for each individual shopper,

4. Includes built-in tools for collecting and reporting user data and events the marketing manager could test how it plays,
5. Implements sales cycle.

Presale services perform the following specific tasks:

1. Supply descriptions and cross-reference information on products offered for sale,
2. Check the availability of the items ordered and report whether they are in stock, back-ordered, and so on.
3. Determine prices for the items a shopper selects, applying promotional sale prices, member discounts, or other factors as appropriate
4. Calculate the cost of shipping and handling for items selected
5. Calculate the tax due.

Sale services include the following:

1. Order capture: Processing customers' orders, totalling prices, authorizing and accepting payment,
2. Order routing: Routing orders to the appropriate service providers and delivery carriers,
3. Postsale services primarily involve order and customer status, enabling both the retailer and the customer to check, audit, and track account and order status after a purchase.

Provides the following management tools (all tools are Web-based and none requires knowledge of programming or HTML):

1. Product management pages: for laying out the products in an online store (Department and Product pages) and adjusting information, including pricing information, while the store is open,
2. Order management pages: for viewing orders sorted by day, month, product, or shopper,

3. Shopper management pages: for viewing information about shoppers,
4. Promotion management pages: for establishing merchandise promotions such as item.

Chapter 9

Results and Conclusions

9.1.Results:

The system was finally executed to validate its functionality. The system is successfully retrieving the data from the databases namely product table, customer table. Successfully throwing a form when the button for registration is pressed. The data entered in the form is validated and ensured that only correct data is fed. Pressing the submit button is pressed, the data is successfully written into customer table after duly checking that there is on duplicity i.e. no two customers with same name and e-card number. On choosing the order placement option, it throws an order form displaying the products available, price per unit and provides text boxes for entering required quantity for each item. It successfully computes the total amount based on the choice of quantity for each item which it reads from form as and when data is filled. These computations are done at the client side and displayed.

The system itself generates the date and also order number. The details date, order number, productwise quantity, total amount along with customer Id number and e-card number.

The results are enclosed in the appendix. The results shown were saved as text files and as result much of the content, which would have been present, had they were saved as HTML pages, lost. However, the data fetched from product table is saved, as HTML page is how each screen appears.

The other screens like welcome page, screens showing the messages were not included in the appendix. The enterprise that adapts this package need not update it's databases through SQL statements. The system contains a client program named manager .By running the program at host site, all the databases can be read and can be updated.

9.2.Conclusion:

The application is a full-fledged application suitable for a small business enterprise to adapt for online business. The package permits the business house to display its range of products, allow remote customers to place orders. It accepts payments through e-card issued by one bank. So far it is good.

But this package has not implemented security layer, has no provision for establishing communication channel with multiple bankers, doesn't address issues like Supply Chain Mechanism, B2B, B2C etc. These are the functional constraints with the package.

The data storehouse is also less exhaustive. But with all these constraints, to a small business enterprise it has sufficient functionality to conduct online business on the net.

The user screens provided are colourless and bare HTML pages. Images, sound and features can be added to make the screens attractive and the user's interest is sustained. But it is not do any thing with the system functionality. Hence minor aberration

9.3.Future Enhancements:

It is mentioned already in the conclusion that certain features are not incorporated in the package. Certain enhancements like creating vast data storehouse to take care of storage requirements, specifying the payment modes and delivery mechanism needs are not addressed by the system. Hence the system is called as Online Webshopping. The system is such that an enterprise, a small business house that can not afford the investment requirements of a full-fledged e-commerce application, can adapt and host it on the net. It is putting up a store on the net and allowing customers to place orders on line.

However, the system fails to address certain problems like security, storing of each and every detail of the entire transactions etc. These are necessary to make it robust.

Apart from the above enhancements required to make it a good system, the system can be developed such that it is in itself a complete e-commerce application. This means addressing a whole lot of issues. Issues like interaction with all the

Bibliography

1.Java 2 for Platform	Joseph Webber QUE Publications
2.The Complete Reference Java 1.2	Patrick Naughton
3.Thinking in Java	Bruce Heckle
4.Java Network Programming	Patrick Naughton
5.The Complete Reference – Java 2	P.Naughton & Herbert Shieldt
6.JDBC Access with Java	Gralam Hamilton, Rick Cattel & Mayden Fisher
7.Jaba Class libraries	Patrick Chan & Rosonna Lee
8.Sevlets Programming	O'Reilly Publications
9.Advanced Javascript programming	Shralin & Shralin.

Webservers

Visitors visiting a site will see what is presented to them. But the developer had too many options. A baseline must be decided so that an average browser can comfortably present the data.

Webserver software meant for Unix will not run on Windows 95 or 98 and vice versa. Selection of Webserver software largely depends on the choice of Operating Systems.

Here is the list of few servers:

Unix

NCSA (Very old)

CERN

Apache server

Windows NT

Internet Information Server(comes with Nt 4.0)

Sambar sever

Netscape's Webmaster

Windows 95

The ideal server is Sambar

Apart from the above, a host of Webserver are available, many free of cost.

Sun's Java Webserver

W3C's Jigsaw

O'Reilly's Website Professional

Lotus's Domino Go Webserver

And

Weblogic.

Appendix A

Results

1. Welcome page:

WELCOME TO WEBSHOPING

EB SHOPPING

NEXT Status(); SCROLL_FUN();

Screen for customer choose options

Result1. Result screen when customer chooses the option 'customers'

LIST OF CUSTOMERS

CID	CNAME	CADDRESS	CCREDITCARDNO
1001	ram	gandhinagar	hm2003
1002	murali	varasiguda	hm2002
1003	harimurthy	warangal	hm2001
1004	prem	nallakunta	hm2004
1005	as	as	hm2001
1006	ajau	Null	hm2001
1007	prem	Null	hm2001
1008	harimurthy	Null	hm2001

To Register

Result2. Order placement form is displayed for the customer who_was registered and logged in properly. Saved as plain text file & not as HTML

```
        Registered Customer, enter Details

===                          Top of Form 1
        Name:
        Creditcard :
        Password   :

        Customer Id:


        Bottom of Form 1
```

Result3: Saved as text file and the tabular display lost.

```
<!-- To check the Id,orders.htm  -->function
```

Order placement form

```
ORDER FORM

  ORDER NO

  ORDER DATE

  PRODUCTS NAME     :       PRICE      :   QUANTITY

  pc
  tv
  musicsystem
  wishingmachine
  refrigerator
              TOTAL AMOUNT

              CUSTOMER ID
```

Result4. Product details–file saved as text file

```
LIST OF PRODUCTS

 PID  PNAME            PSTOCK PRATEPERUNIT
 1    pc              5      10000
 2    tv              10     12000
 3    musicsystem     8      20000
 4    wishingmachine  6      7000
 5    refrigerator    10     7000
```

Result5. Customer registration form

CUSTOMER Id : 1009

CUSTOMER NAME :

CUSTOMER PASSWORD :

CUSTOMER CREDITCARD NO :

CUSTOMER ADDR :

REGISTER CLEAR