

1. INTRODUCTION

1.1 ORGANISATION PROFILE

Hindustan software limited established in 1995, has been actively involved in development of customized software for various clients since its inception and also has its presence strongly felt in the areas of Training & consulting. Market Leadership through customer satisfaction, a commitment to excellence & high growth rate, these have characterized HSL in its rapid climb in Information Technology Industry.

Development

Our software development team strives to develop innovative software that meets customer needs. A team of developers with exclusive educational background and excellent computing skills exert their maximum efforts and endeavor to satisfy the specialized requirement of our esteemed clients. Our expertise and vast exposure qualifies us in developing software's in Information systems, Web Based Applications, Internet Solution, Web page Design and Hosting. Our team consists of expertise in Java, VC++, VB, Oracle and all Internet Application Development Tools.

Training

Hindustan has trained more than 1000 professionals so far in the area of JAVA and VC++. All the students have launched in a very good career path both in India & in abroad. The curriculum matches latest trends and make the students to compete in the IT industry.

Consulting

The training division gives us an opportunity to meet variety of candidates with various skill sets. Our database consists of details regarding more than 1000 professionals. So we are fit to assist all sort of IT Human resource consultancies and IT companies in India and abroad.

Excellence through “TEAM WORK”-The Philosophy of Total Quality Management is inculcated in every HSL employee through intensive training programs. Our investment on a highly skilled and motivated manpower constitutes towards establishing HSL as a truly quality conscious company, continually striving to bring the finest solutions to the discerning customers.

1.2 ABOUT THE PROJECT

DISTRIBUTOR MANAGEMENT SYSTEM is a web based application, which is developed for a particular company for maintaining and analyzing the sales of the product. This project is a module under website development for a particular computer hardware company. Here all communications held through internet and its database has all the updated information of the sales details.

The main objective of the project is to analyze the sales of the products by a manager through the details supplied by the distributors, sales managers and representatives. It is very useful for the distributors, sales managers to know about the sales of the products done by them and by others in particular area/zone.

This system gives complete analysis about the moving of the product in the market and the person responsible for selling the product in that area. It is computerized to improve the efficiency of the organization by reducing the cost of maintaining data and minimizing the time involved in handling the data.

Depending on the access rights given the users can process different modules.

The modules are as follows

- ☐ Administrative department
- ☐ Manager department
- ☐ Distributors department
- ☐ Sales Managers department &
- ☐ Representatives department

2. ABSTRACT/SYNOPSIS

DISTRIBUTOR MANAGEMENT SYSTEM is a web based application, which is developed for a particular company for maintaining and analyzing the sales of the product. This project is a module under website development for a particular computer hardware company. Here all communications held through internet and its database has all the updated information of the sales details. Depending on the access rights given the users can process different modules.

The modules are as follows

- ☐ Administrative department
- ☐ Manager department
- ☐ Distributors department
- ☐ Sales Managers department &
- ☐ Representatives department

Each module controls the lower modules with their given rights.

The main objective of the project is to analyze the sales of the products by a manager through the details supplied by the distributors, sales managers and representatives. It is very useful for the distributors, sales managers to know about the sales of the products done by them and by others in particular area/zone.

This system gives complete analysis about the moving of the product in the market and the person responsible for selling the product in that area. It is computerized to improve the efficiency of the organization by reducing the cost of maintaining data and minimizing the time involved in handling the data.

Administrative department is the department which has veto rights in this project. This department has the right to remove/add any persons from/to the system or can change any product details from the database and can send mail to any person in the system.

Managers department is the department made to assist and to reduce the work load to the administrative department. This department has no rights to change product details or add/remove persons to/from the system. This department sets the targets to the distributors department in monthly basics and to evaluate the distributors.

Distributors department has distributors as its members. Each distributor has to submit their sales daily via, a sales form. The distributor's sets the target to the sales managers, in weekly basics and evaluates their progress daily. The distributors cannot send mail to the administrative department.

Sales Managers department has sales managers as its members. Each sales manager has to submit their sales daily via, a sales form. The sales manager sets the target to the sales representative's, in daily basics and evaluates their progress daily. The sales managers cannot send mail to the administrative department nor to Managers department.

Sales Representative's department has sales representative's as its members. Each sales representative's, has to submit their sales daily via, a sales form. The sales representative's, can send mail only to the Sales Managers department.

3. SYSTEM ANALYSIS

System analysis focuses on specifying what the system or application required to do. It allows individuals see logical elements (what the system should do) apart from the physical component it uses (computers, terminals and storage system). It is the process of gathering and interpreting facts, diagnosing problem and using the information to recommend improvements to the system.

3.1 Existing System:

The existing system is the manual system. The manual system is error prone. It is time consuming. It is very difficult for a person to produce report. There are chances for manipulating the sales reports and changing it. This system involves a lot of manual entries with the applications to perform the desired task.

Limitations:

- ☐ Data maintenance adopted by the present system is not accurate.
- ☐ Inaccurate result in case of duplicating, delay and inconsistency in reporting.

- ☐ Generating consolidated reports is more difficult in manual system and it may not be consistent.
- ☐ The transactions are very time consuming.
- ☐ There is no facility for the users to know whether the data is entered is valid or not. This disadvantage is the major cause of errors in transaction.
- ☐ There is inconsistency in maintaining data's.
- ☐ No global view ability of data's.
- ☐ No chance of knowing other members sales daily.
- ☐ It is not user friendly.

3.2 Proposed System:

The proposed system is designed to eliminate the drawbacks of the existing system. It is designed by keeping in mind all the drawbacks of the present system in order to provide a permanent solution to the problems. The primary aim of the new system is to speedup transactions. The report is prepared for the sales done by the representative and the distributor.

The representative code and distributor code are validated. Accuracy for all the data entered is maintained in the proposed system through validation and verification from all the files. Verification of representative code, distributor code, validation of records, etc., are performed with maximum accuracy. In short, efficiency is established. The advantages of this system are

- ☐ User friendliness is the keyword for all the new software in the market. The proposed system incorporates this concept into itself to guide the user.

- At every stage of data entry necessary comments and validation messages are provided to the user.
- The proposed system is also expected to reduce the amount of paper work involved. The hard copies of only necessary documents need to be taken the rest can be avoided.
- Competitive environment is developed with the publications of other members of same level.
- Good communication is being held via mail options.
- Data's are maintained accurately via daily submission of sales details.

3.3 Need For Computerization:

The Project entitled "Distributor management System" was not computerized previously. In this system, the information about the sales and the progress of sales and the movement of the products in the market cannot be noted easily. If these information's are needed we need to spend lot of time, money and energy.

If the company's turn over has to be increased then there is some need for experts ideas for that up to date information has to supplied this is quite impossible in the present system.

If the company wants to inform some important meeting/decisions, this process should start before sometime before the actual happening. If some products are added this too has to be informed individually for that quite some money has to be spent.

Reports cannot be created easily with much flexibility and with limited amount of time. The head of the company cannot sit at a place and maintain the sales records without much assistance.

Data maintenance is quite tough when maintained manually. High level of data redundancy exists.

3.4 Packages Selected:

Front End	: Java Servlets
Web Tools	: HTML
Web Server	: Java Web Server2.0
Validation	: JavaScripts
Backend	:MS-Access

3.5 Resource Required:

Processor	: Pentium-s II 300 mhz
RAM	: 64 MB
HARDDISK	: 6 GB
KEYBOARD	: 108

MOUSE	: LOGITECH
FLOPPY DISK	: 1.44MB
DISPLAY TYPE	: EGA/VGA
SERIAL PORTS	: 2F8/3F8
PARALLEL PORTS	: 15 (if need)
CACHE MEMORY	: 512K
BASE MEMORY	: 640K
EXTENDED MEMORY	:31744K
L2 CACHE TYPE	:PIPELINE

3.6 Feasibility Study

Feasibility study is the measure of how beneficial, the development of information system would be to an organization, and it is mainly Conducted

- To determine whether this is a new and better way to do the job that will be benefiting the end user.
- To ascertain the cost and savings of alternatives.
- To recommend the best of alternatives.

Unfortunately the development of a computer-based system more likely to be plagued by a scarcity of resources difficulty delivery dates. It is both necessary and prudent to evaluate the feasibility of a project at the earliest possible time. When we talk about feasibility study we concentrate our attention on four primary areas of interest:

1. Economic Feasibility

Economic feasibility means an evaluation of development cost weighed against the ultimate income or benefit from the developed system. The benefits of the system to users outweigh the costs incurred during the system

development. In addition, flexibility of the system to incorporate further enhancements improves the performance to suit the future needs of the user.

2. Technical Feasibility

The technical feasibility involves the study of the Hardware and software requirements and if it is feasible to obtain the required parameters.

3. Operational Feasibility

The Proposed project is beneficial because it can be turned into a website that will meet the user's requirements. Since the request has been from the administrator As well as from the users, it does not have any operational barriers. So it is operationally feasible.

3.7 Analysis of proposed system:

While analyzing the proposed system, we should know the users recruitments and whether the system needs these recruitments. The user can easily enter and retrieve the all-available details. The system was developed by the user interaction manner. The calculation and validation should be free from error and reports.

Advantage over existing system:

This system provides advantage like:

- Data redundancy is avoided
- Data integrity is maintained
- Security is ensured
- Report can be easily generated so this system feasibility.

3.8 Normalization:

Normalization is nothing but in the database system we have tables we may have to delete a record from the tables or update or modify from the tables. But some times when we are trying to do those operations we may get the incorrect answer or result or some error. The main reason for that table is not normalized. Therefore Normalization is very important to develop the software. Every software engineer must do the normalization before developing the software. Therefore in this developed software the normalization is done in order to avoid such a problem and redundancy of the records.

First Normal Form

A repeating group that is the reoccurrence of data item or group of record is actually another relation. This can be removed from the record and treated as an additional record structure or relation. A relational model does not permit or support unnormalized tables. A relation scheme is said to be first normal (1NF) if the values in the domain of attribute of the relation are atomic.

Second Normal Form

The second Normal form is achieved whenever the data item in record that is not dependent on the primary key of the record should be removed from the table and used in a separate table. A relation scheme is said to be second normal form (2NF) if it is in the 1NF and if all nonprime attributes are fully functionally dependent on the relation keys. A database scheme is in second normal form if every relation scheme included in the database scheme is in second normal form.

Third Normal Form

A relation Scheme $R\langle S, F \rangle$ is in third normal form (3NF) if for all nontrivial functional dependencies in F^+ of the form $X \rightarrow A$ (X is Super Key). A

table is said to be in third normal form if, it is in second normal form and every non-key attribute is functionally dependent o just the primary key.

4. PROJECT DESCRIPTION

The project "**Distributor management system**" deals with the process of managing the distributors of a company. This project consists of many modules like Administrative department, Manager department, Distributor

department, Sales manager department and Representative department with their access rights as sub modules.

ADMINISTRATIVE DEPARTMENT:

Administrative department module is mainly for the administrator. The administrator monitors the whole sales activities. The administrator is the one with veto rights. The administrator has special rights to dismiss/add any person in this system, to remove/modify the products details, to give target values, to change any ones addresses and to view various reports for monitoring purpose.

Add Distributor/Sales manager/Representative:

The administrator gets the details of the persons to be added via mails from other major modules. The details of Distributor, Sales manager, Representative are added in disp, smp, repp tables respectively with correct userid and position ids.

Removal Distributor/Sales manager/Representative:

The administrator removes Distributor/Sales manager/Representative the system according to their performance. The details of Distributor, Sales manager, Representative are deleted from disp, smp, repp tables respectively by supplying their user ids.

Product details modification:

In this sub module the product detail is modified via the product keys, if the product key has to be changed the product key is send as session value. The products can be added by specifying the required data's. These data's are updated in the protab table according to the process for which it is called.

Mail Options:

In this, the administrator can send mails to any person in the system. The subject of the mails can be memo, target value, incentives, any advice or

any required subject. These data's are stored in mop table with from, to, subject, date as the fields.

MANAGER DEPARTMENT:

Manager department is the module developed for assisting the administrative department. This module has no veto rights, it just evaluates the distributors and send details to the administrative module. The evaluation is stored in the sales, reppo tables for report generation.

DISTRIBUTORS DEPARTMENT:

Distributors department has distributors as its members. This module has the following as the sub modules Mail options, Evaluation and User input form. Each distributor has to submit their sales daily via, a sales form. The distributor's sets the target to the SALES MANAGERS., in weekly basics and evaluates their progress daily. The distributors cannot send mail to the administrative department.

Mail options:

The difference between the administrative department mail options and this mail options is the distributor cannot send mail to the administrative department but mails can be send to other persons in the system.

Evaluation:

This sub module is used for evaluating the sales manger of a distributor on daily basis. This evaluation is done on the comparison of target achieved and the target to be achieved. The distributor supplies the current position of the sales manager and gives comment on the performances; the comment may include incentives also. These evaluations will be stored in the reppo table, since it will be displayed for other sales mangers.

SALES MANAGERS DEPARTMENT:

Sales Managers department has sales managers as its members. This module has the following as the sub modules Mail options, Evaluation and User input form. Each sales manager has to submit their sales daily via, a sales form. The sales managers sets the target to the sales representative's., in daily basics and evaluates their progress daily. The sales managers cannot send mail to the administrative department nor to Managers department.

Mail options:

The difference between the distributors department mail options and this mail options is the sales manager cannot send mail to the administrative department and to manager department but mails can be send to other persons in the system.

Evaluation:

This sub module is used for evaluating the representatives of a sales manger on daily basis. This evaluation is done on the comparison of target achieved and the target to be achieved. The sales manger supplies the current position of the sales manager and gives comment on the performances; the comment may include incentives also. These evaluations will be stored in the reppo table, since it will be displayed for other sales mangers.

REPRESENTATIVE'S DEPARTMENT:

Representative's department has sales representative's as its members. Each sales representative's., has to submit their sales daily via, a sales form. The sales representative's., can send mail only to the Sales Managers department.

5. DATABASE DESCRIPTION

Table Name: logs

Field Name	Type	Constraint
usnm	Text	Unique
pass	Text	Not null
cod	Text	Primary key
pos	Text	Not null

Table Name: protab

Field Name	Type	Constraint
pcod	Integer(11)	Primary key
pnm	Integer(11)	Unique
pcost	Number	Not null

Table Name: disp

Field Name	Type	Constraint
dcod	Text	Primary key
usnm	Text	Unique
pass	Text	Not null
nm	Text	Not null
add1	Number	Not null
cit	Text	Not null
sat	Text	Not null
pin	Text	Not null
email	Text	Not null
smcod1	Number	Not null
smcod2	Text	Not null
area	Text	Unique
target	Text	Not null

Table Name: smp

Field Name	Type	Constraint
dcod	Text	Foreign key
smcod	Text	Primary key
usnm	Text	Unique
pass	Text	Not null
nm	Text	Not null
add1	Text	Not null
cit	Text	Not null
sat	Text	Not null
pin	Number	Not null
email	Text	Not null
r1	Text	Unique
r2	Text	Unique
area	Text	Unique
target	Number	Not null

Table Name: repp

Field Name	Type	Constraint
dcod	Text	Foreign key
smcod	Text	Foreign key
rcod	Text	Primary key
usnm	Text	Not null
pass	Text	Not null
rnm	Text	Not null
dob	Text	Not null
qua	Text	Not null
add	Text	Not null
cit	Text	Not null
sat	Text	Not null
pin	Number	Not null
phn	Number	Not null
email	Text	Not null
area	Text	Unique
target	Number	Not null

Table Name: sales

Field Name	Type	Constraint
cod	Text	Primary key
dat	Text	Not null
pos	Text	Not null
ps1	Number	Not null
ps2	Number	Not null
ps3	Number	Not null

Table Name: mop

Field Name	Type	Constraint
cod	Text	Primary key
from	Text	Not null
to	Text	Not null
sub	Text	Not null
mes	Text	Not null
dat	Date	Not null

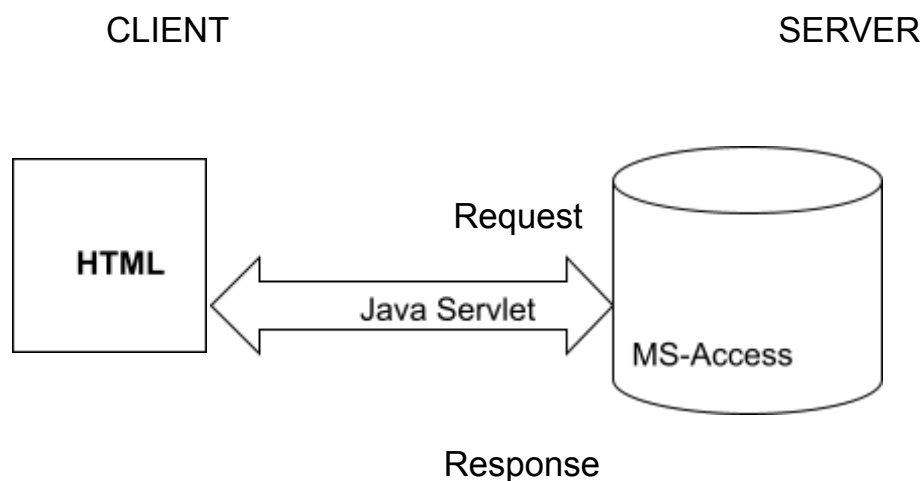
Table Name: repo

Field Name	Type	Constraint
cod	Text	Primarykey
totsalpr	Text	Not null
posi	Text	Not null
pos	Text	Not null
comt	Text	Not null
dat	Date	Not null

6. SOFTWARE DESCRIPTION

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 you use to write servlets, assumes nothing about the server's environment or protocol. Servlets have become most widely used within HTTP servers; many web servers support Java Servlet technology.

Use 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.

Other Uses for Servlets

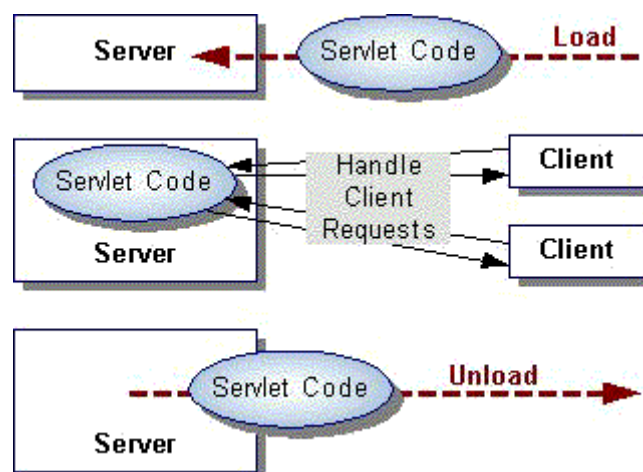
Here are a few of the many applications for servlets:

- Allowing collaboration between people. A servlet can handle multiple requests concurrently, and can synchronize requests. This allows servlets to support systems such as on-line conferencing.
- 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 organizational boundaries.

The Servlet Life Cycle

Each servlet has the same life cycle:

- A server loads and initializes the servlet
 - The servlet handles zero or more client requests
 - The server removes the servlet
- (some servers do this step only when they shut down)



Initializing a Servlet

When a server loads a servlet, the server runs the servlet's init method. Initialization completes before client requests are handled and before the servlet is destroyed.

Even though most servlets are run in multi-threaded servers, servlets have no concurrency issues during servlet initialization. The server calls the init method once, when the server loads the servlet, and will not call the init method again unless the server is reloading the servlet. The server can not

reload a servlet until after the server has destroyed the servlet by running the destroy method.

Destroying a Servlet

Servlets run until the server destroys them, for example, at the request of a system administrator. When a server destroys a servlet, the server runs the servlet's destroy method. The method is run once; the server will not run the destroy method again until after the server reloads and reinitializes the servlet.

When the server calls the destroy method, another thread might be running a service request. The Handling Service Threads at Servlet Termination lesson shows you how to provide a clean shutdown when there could be long-running threads still running service requests.

Saving Client State:

Saving Client State shows you how to use session tracking and cookies.

The Servlet API provides two ways to track client state:

Session Tracking

Session tracking is a mechanism that servlets use to maintain state about a series of requests from the same user (that is, requests originating from the same browser) across some period of time.

Sessions are shared among the servlets accessed by a client. This is convenient for applications made up of multiple All the servlets in the example have access to the user's session.

To use session tracking,

- Get a session (an `Http Session` object) for a user.
- Store or get data from the `HttpSession` object.
- Invalidate the session (optional).

Cookies

Cookies are a way for a server (or a servlet, as part of a server) to send some information to a client to store, and for the server to later retrieve its data from that client. Servlets send cookies to clients by adding fields to HTTP response headers. Clients automatically return cookies by adding fields to HTTP request headers.

Each HTTP request and response header is named and has a single value. For example, a cookie could be a header named `BookToBuy` with a value `304qty1`, indicating to the calling application that the user wants to buy one copy of the book with stock number 304. (Cookies and their values are application-specific.)

Multiple cookies can have the same name. In addition to a name and a value, you can also provide optional attributes such as comments. Current web browsers do not always treat the optional attributes correctly, so you should not rely on them.

A server can provide one or more cookies to a client. Client software, such as a web browser, is expected to support twenty cookies per host, of at least four kilobytes each

When you send a cookie to a client, standard HTTP/1.0 caches will not cache the page. Currently, the `javax.servlet.http.Cookie` does not support HTTP/1.1 cache control models.

To send a cookie,

1. Instantiate a Cookie object
2. Set any attributes
3. Send the cookie

To get information from a cookie,

1. Retrieve all the cookies from the user's request
2. Find the cookie or cookies with the name that you are interested in, using standard programming techniques
3. Get the values of the cookies that you found

JDBC

JDBC™ was designed to keep simple things simple. This means that the JDBC API makes everyday database tasks, like simple SELECT statements, very easy.

Establishing a Connection

The first thing you need to do is establish a connection with the DBMS you want to use. This involves two steps: (1) loading the driver and (2) making the connection.

Loading Drivers

Loading the driver or drivers you want to use is very simple and involves just one line of code. If, for example, you want to use the JDBC-ODBC Bridge driver, the following code will load it:

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

Your driver documentation will give you the class name to use. For instance, if the class name is `jdbc.DriverXYZ`, you would load the driver with the following line of code:

```
Class.forName("jdbc.DriverXYZ");
```

You do not need to create an instance of a driver and register it with the `DriverManager` because calling `Class.forName` will do that for you automatically. If you were to create your own instance, you would be creating an unnecessary duplicate, but it would do no harm.

When you have loaded a driver, it is available for making a connection with a DBMS.

Making the Connection

The second step in establishing a connection is to have the appropriate driver connect to the DBMS. The following line of code illustrates the general idea:

```
Connection con = DriverManager.getConnection(url,  
                                             "myLogin", "myPassword");
```

This step is also simple, with the hardest thing being what to supply for `url`. If you are using the JDBC-ODBC Bridge driver, the JDBC URL will start with `jdbc:odbc:`. The rest of the URL is generally your data source name or database system. So, if you are using ODBC to access an ODBC data source called "Fred", for example, your JDBC URL could be `jdbc:odbc:Fred`. In place of "myLogin" you put the name you use to log in to the DBMS; in place of "myPassword" you put your password for the DBMS. So if you log in to

your DBMS with a login name of " Fernanda " and a password of " J8, " just these two lines of code will establish a connection:

```
String url = "jdbc:odbc:Fred";  
Connection con = DriverManager.getConnection(url, "Fernanda", "J8");
```

If you are using a JDBC driver developed by a third party, the documentation will tell you what subprotocol to use, that is, what to put after jdbc: in the JDBC URL. For example, if the driver developer has registered the name acme as the subprotocol, the first and second parts of the JDBC URL will be jdbc:acme: . The driver documentation will also give you guidelines for the rest of the JDBC URL. This last part of the JDBC URL supplies information for identifying the data source.

If one of the drivers you loaded recognizes the JDBC URL supplied to the method `DriverManager.getConnection`, that driver will establish a connection to the DBMS specified in the JDBC URL. The `DriverManager` class, true to its name, manages all of the details of establishing the connection for you behind the scenes. Unless you are writing a driver, you will probably never use any of the methods in the interface `Driver` , and the only `DriverManager` method you really need to know is `DriverManager.getConnection` .

The connection returned by the method `DriverManager.getConnection` is an open connection you can use to create JDBC statements that pass your SQL statements to the DBMS. In the previous example, `con` is an open connection.

JavaScript:

JavaScript is an open scripting language that anyone can use without purchasing a license developed by Netscape. JavaScript may be considered a derivative of the programming language Java. Both are tools for providing interactivity into web pages.

- ☐ A JavaScript is lines of executable computer code.
- ☐ A JavaScript can be inserted into an HTML page.
- ☐ JavaScript is supported by all major browsers like Netscape and Internet Explorer.
- ☐ JavaScript can be used to validate data.
- ☐ JavaScript is a set of programming commands and instructions that can be used to enhance the way the page looks, insert or delete parts of a page's contents depending on system requirements, control the operation of Web Server, communicate with a online database or perform any task previously associated with CGI communications.

MICROSOFT ACCESS

Using Microsoft Access, we can manage all user information from a single database file. Within the file, divide the data into separate storage containers called tables, view, add and update table using online forms, find and retrieve just the data we want using queries, and analyze or print data in a specific order using reports.

To want where data, create one table for each type of information we track.

To bring the data from multiple tables together in a query, form, or, a report, we define relationship between the tables.

To find and retrieve just the data that meets conditions we specify, including data from multiple tables, create a query. A query can also update or delete multiple records at the same time, and perform built-in or custom calculation on data.

To easily view, enter and change data directly in a table, create a form. When we open a form, Microsoft access retrieves the data from one or more tables and displays it on the screen using the layout we choose in the form wizard or using a layout that we created from scratch.

The following drag-drop functionality is now available in Microsoft Access:

- ☐ We can drag and drop database objects between open Microsoft Access databases.
- ☐ We can drag and drop Microsoft Access tables and queries into other applications, such as Microsoft word and Excel.
- ☐ We can create a table by dragging and dropping a range of cells from Microsoft Access Excel Spreadsheet into Database window.

Microsoft Access provides extensive features designed to help easily the Internet and develop a World Web Application. You need a web browser, such as Microsoft Internet Explorer, modem, Internet connection or other network connection to access the Internet and take advantage of some of these new features.

- ☐ Securing and administering a database.
- ☐ Hide database objects that you don't want other user to see or open database window.
- ☐ Assign a single password to control that can open a database instead of or in addition to implementing user level security.
- ☐ Create a query.

Microsoft Access can often create a query for you so you don't have to design one from scratch.

- To create a query use the basis of a form or report, try the form or report wizards. They create the form or reports, and if it's based on more than one table; they also create its underlying SQL statements as a query.
- To easily create queries that you want to run independently based on multiple forms and reports, try one on query wizards. Query wizard do all the basic work for you after you provide answers to a series of questions. Even if you have created many queries you may want to use a wizard to quickly design the query. Then you can switch to design view to customize it.
- To create queries from filters you created using filter by form, Filter by selection or filter for input, save filter as a query.

If none of these methods satisfies your needs, you can create the query from scratch in query design view.

Java Web Server

Sun's Java Web Server(JWS) was the first commercial HTTP server to support servlets and should receive strong consideration when selecting a Java-enabled Web Server.It is used to register and invoke servlets using JWS.

ARCHITECTURAL DESIGN

Dynamic web applications are presented as a three-tier architecture, in which the Java servlets play a key role in the business logic layer of the architecture. Designing the application in layers, or tiers, is useful for many different reasons. In a multiple tier design, each tier can be run a separate

machine, or machines, allowing for improved processing performance. Depending on the design, multiprocessor machines, or many different independent computers can be used to improve performance. Efficient layering can give structure to your application, promote scalability, and ease long-term maintenance requirements for your code.

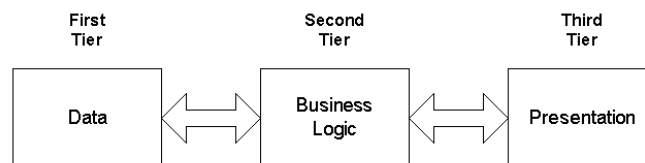


Figure 5.1.1 Three Tier Architecture

First-Tier: *Persistent Data Storage* – The underlying technology to implement this layer could be a variety of things including cookies, server side files or databases.

Second-Tier: *Business Logic Layer* – This is where one code any rules regarding the data stored and generally is the bulk of the application.

Third-Tier: *Presentation* – Enables the user to see the results of the business logic applied to the data stored.

HTML

HTML developed a few years ago as a subset of *SGML* (*Standard Generalized Mark-up Language*) which is a higher-level mark-up language that has long been a favorite of the Department of Defense. Like HTML, it describes formatting and hypertext links, and it defines different components of a document. HTML is definitely the simpler of the two, and although they are related, there are few browsers that support both.

Because HTML was conceived for transmission over the Internet (in the form of Web pages), it is much simpler than SGML, which is more of an application-oriented document format.

While it's true that many programs can load, edit, create, and save files in the SGML format (just as many programs can create and save programs in the Microsoft Word format), SGML is not exactly ideal for transmission across the Internet to many different types of computers, users, and browser applications. HTML is more suited to this task. Designed with these considerations in mind, HTML lets you, the designer, create pages that you are reasonably sure can be read by the entire population of the Web. Even users who are unable to view your graphics, for instance, can experience the bulk of what you're communicating if you design your HTML pages properly.

At the same time, HTML is a simple enough format (at least currently) that typical computer users can generate HTML documents without the benefit of a special application. Creating a WordPerfect-format document would be rather difficult by hand (including all of the required text size, fonts, page breaks, columns, margins, and other information), even if it weren't a "proprietary"-that is, nonpublic-document format.

HTML is a public standard, and simple enough that you can get through a book like this one and have a very strong ability to create HTML documents from scratch. This simplicity is part of a trade-off, as HTML-format documents don't offer nearly the precision of control or depth of formatting options that a WordPerfect- or Adobe PageMaker-formatted document would.

7. DATA DICTIONARY

The data dictionary is an organized listing of all data elements that are pertinent to the system, with precise, rigorous definitions so that both user and system analyst will have a common understanding of inputs, outputs, components of stores and even intermediate calculations.

Name

The primary name of the data or control item, the data store or an external entity. Alias--other names used for the first entry.

Where-used/how-used

A listing of the processes that use the data or control item and how it is used (e.g., input to the process, output from the process, as a store, as an external entity).

Content-description

A notation for representing content. Supplementary information--other information about data-types, preset values (if known), restrictions or limitations and so forth.

This system uses several data elements for identifying the users. Here the user name and password is the primary aspect such that each user is identified, beyond that his email-id, address, phone no etc are used for future reference.

Importances of data dictionary are as follows:

- ☐ To manage the details
- ☐ Communicate meaning
- ☐ Document System features
- ☐ Facilitate analysis
- ☐ Locate errors and omissions

logs

USNM	: user name
PASS	: user password
COD	: user identity
POS	: group identity

protab

PCOD	: product code
PNM	: product name
PCOST	: product cost

disp

DCOD	: distributor's code
USNM	: distributor's name
PASS	: password
NM	: distributor's name
ADD1	: distributor's address
CIT	: distributor's city
SAT	: distributor's state
PIN	: pin code of user
EMAIL	: email-id of user
SMCOD1	: distributor's sales manger code
SMCOD2	: distributor's sales manger code
AREA	: distributor's area of control
TARGET	: distributor's monthly target

smp

DCOD	: distributor's code
SMCOD	: sales manager's code
USNM	: sales managers name

PASS	: sales manager's password
NM	: sales managers name
ADD1	: sales managers address
CIT	: sales manager's city
SAT	: sales manager's state
PIN	: sales manager's pin code
EMAIL	: sales manager's email-id
R1	: sales manager's sales representatives
R2	: sales manager's sales representatives
AREA	: sales manager's area of control
TARGET	: sales manager's weekly target

repp

DCOD	: distributor's code
SMCOD	: sales manager's code
RCOD	: sales representative's code
USNM	: sales representative's name
PASS	: sales representative's password
RNM	: sales representative's name
DOB	: sales representative's date of birth
QUA	: sales representative's qualification
ADD	: sales representative's address
CIT	: sales representative's city
SAT	: sales representative's state
PIN	: sales representative's pin code
EMAIL	: sales representative's email-id
AREA	: sales representative's area of control
TARGET	: sales representative's daily target

sales

COD	: user's code
DAT	: sales date
POS	: user's position
PS1	: number of product of first product sold
PS2	: number of product of second product sold
PS3	: number of product of third product sold

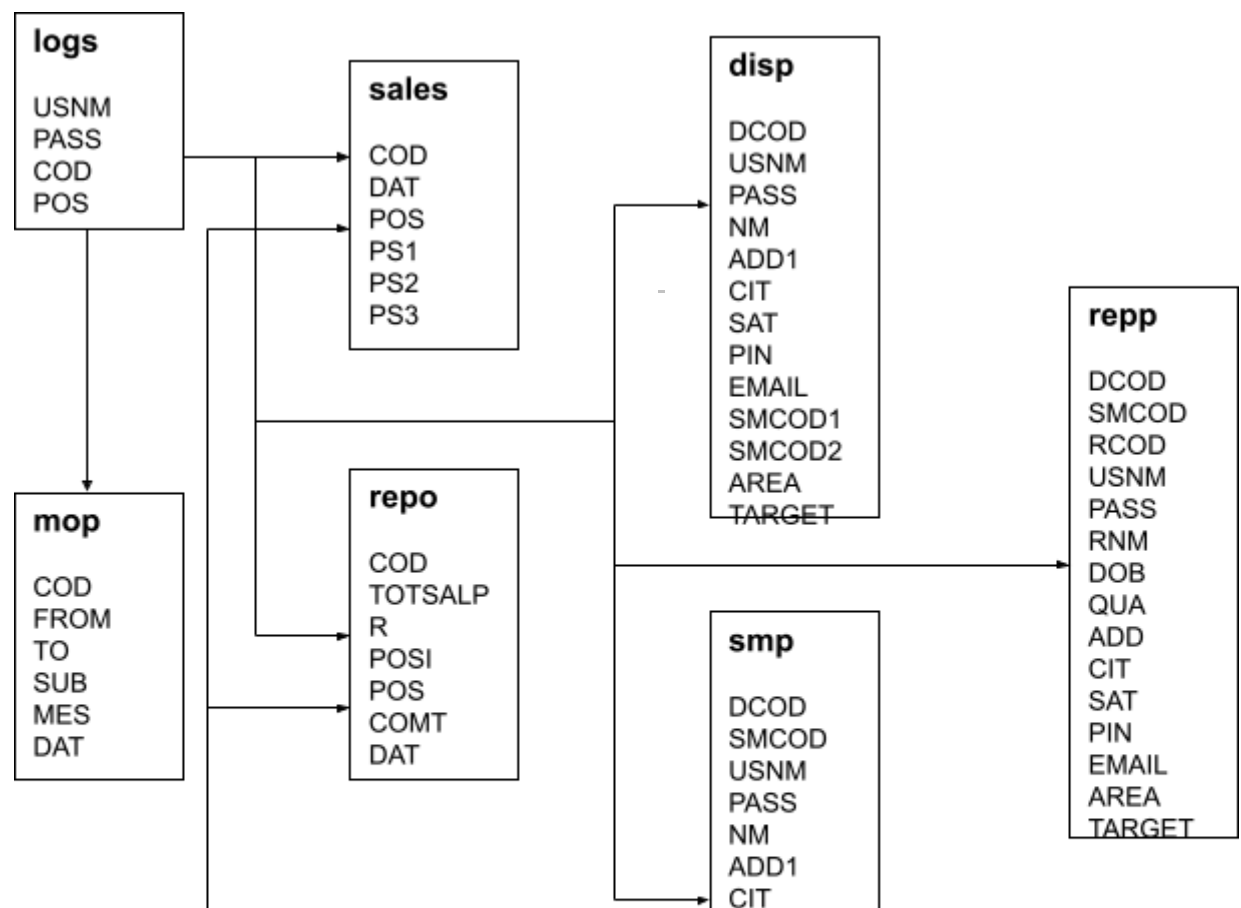
mop

COD	: user's code
FROM	: mail sender's name
TO	: mail receiver's name
SUB	: subject of the mail
MES	: message of the mail
DAT	: date of the mail

repo

COD	: user's code
TOTSALPR	: total sales by the user current day
POSI	: position of the users
POS	: group identity
COMT	: comments of higher authorities
DAT	: date of report

8. TABLE RELATIONSHIP



9. DATA FLOW DIAGRAM

A data flow diagram is a graphical representation that depicts information flow and the transforms that are applied as data move from input to output. The data flow diagram may be used to represent a system or software at any level of abstraction.

In fact, DFD's may be partitioned into levels that represent increasing information flow and functional detail. Therefore DFD provides a mechanism for functional modeling as well as information flow modeling.

A level 0 DFD also called a fundamental system model or a context model, representing the entire software element as a single bubble with input and output data indicated by incoming and outgoing arrows, respectively. Additional processes and information flow paths are represented as the level 0

DFD is partitioned to reveal more detail. For example, a level 1 DFD might contain five or six bubbles with interconnecting arrows. Each of the processes represented at level 1 is a sub function of the overall system depicted in the context model.

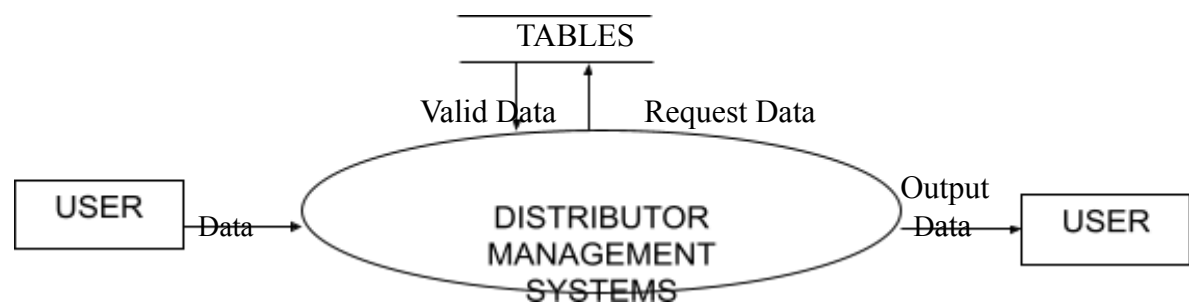
A DFD has the purpose of clarifying system requirements and identifying major transformation that will become programs in system design.

DFD Symbols:

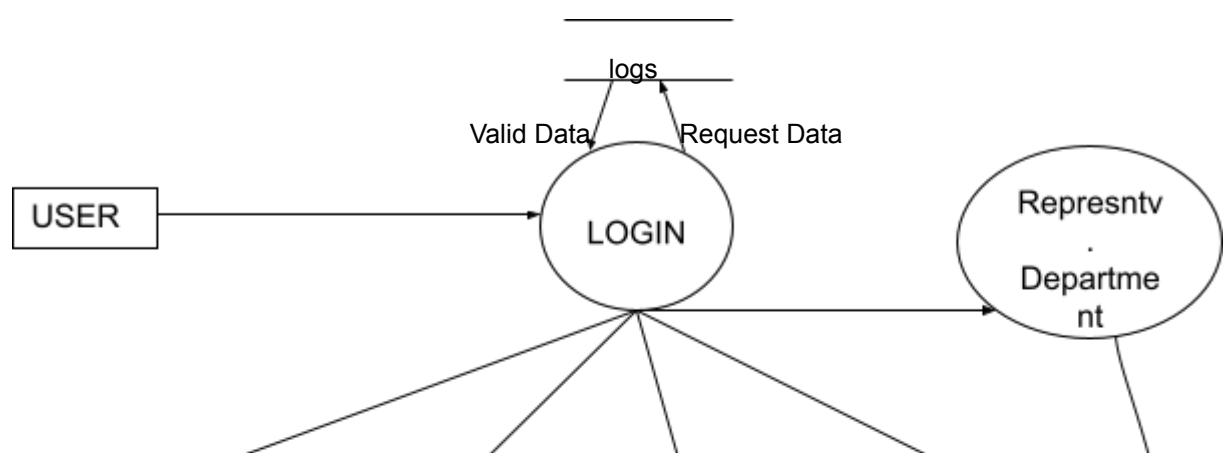
In the DFD there are four symbols.

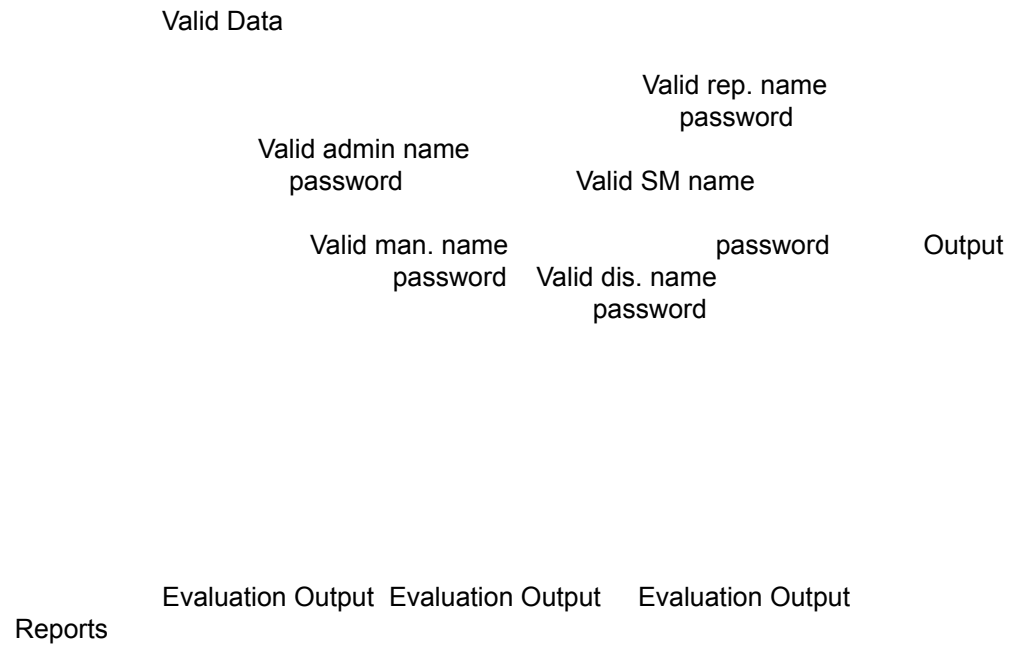
- 1) A square defines a source or destination of system data.
- 2) An arrow identifies data-flow data in motion. It is a pipeline through which information flow.
- 3) A Circle represents a process that transforms incoming data flows into outgoing data flow(s).
- 4) An open rectangle is a data store — data at rest or a temporary repository of data.

O' Level

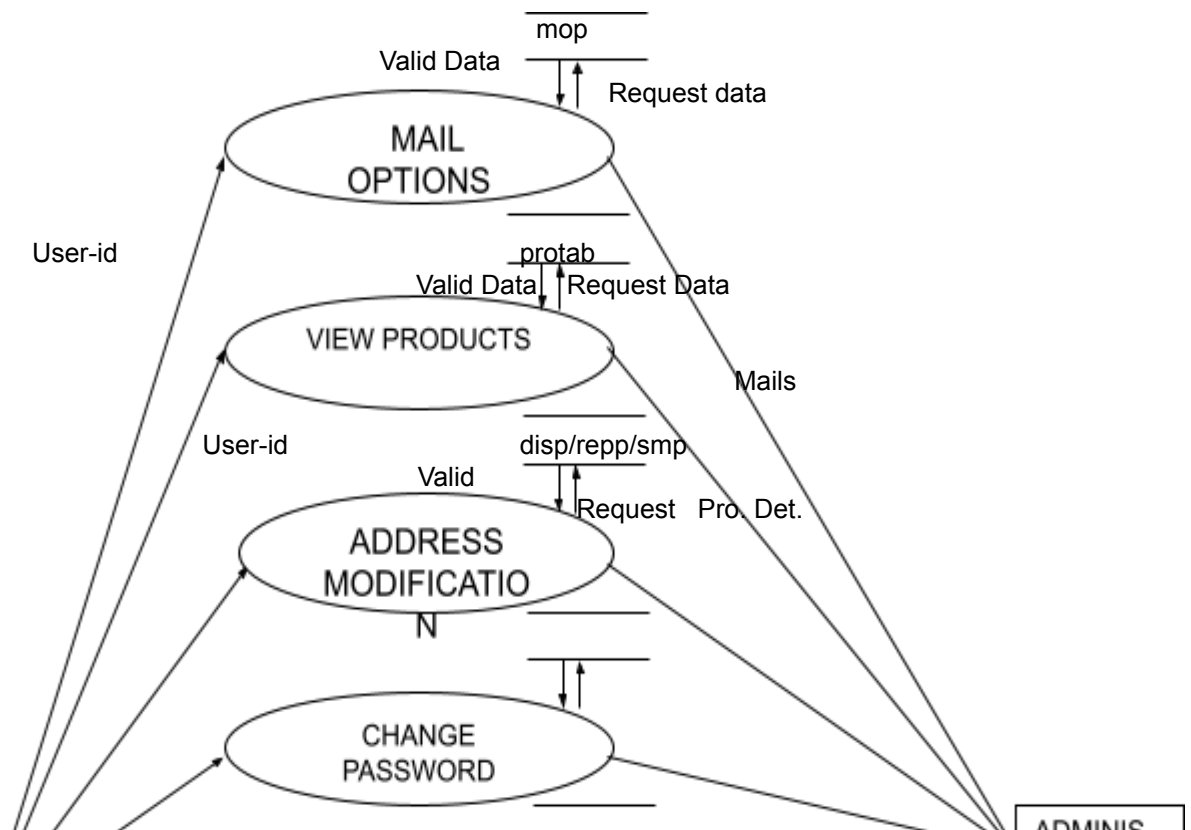


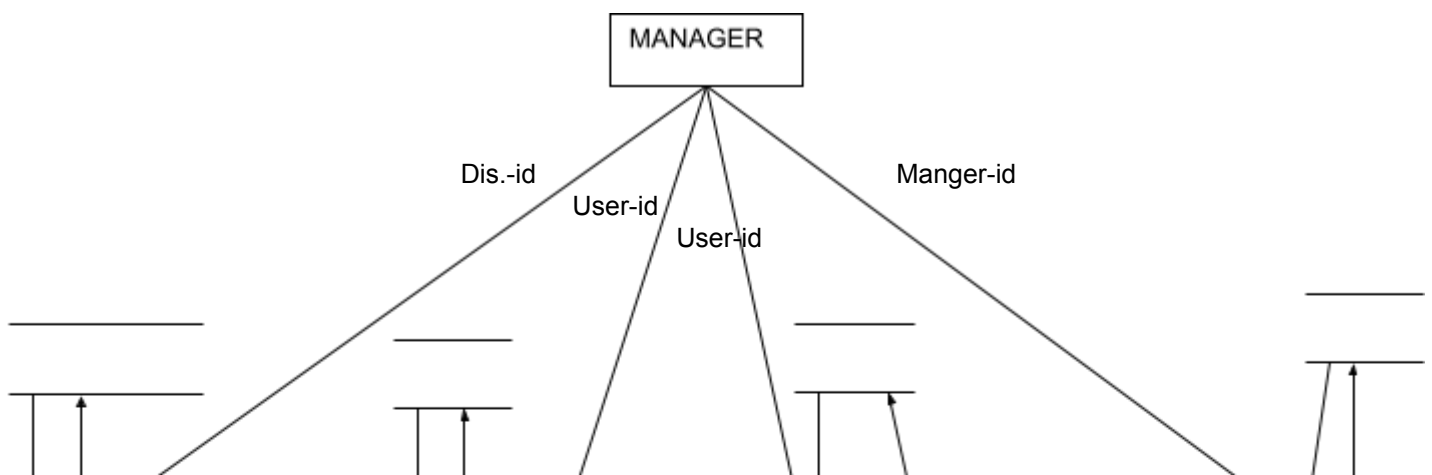
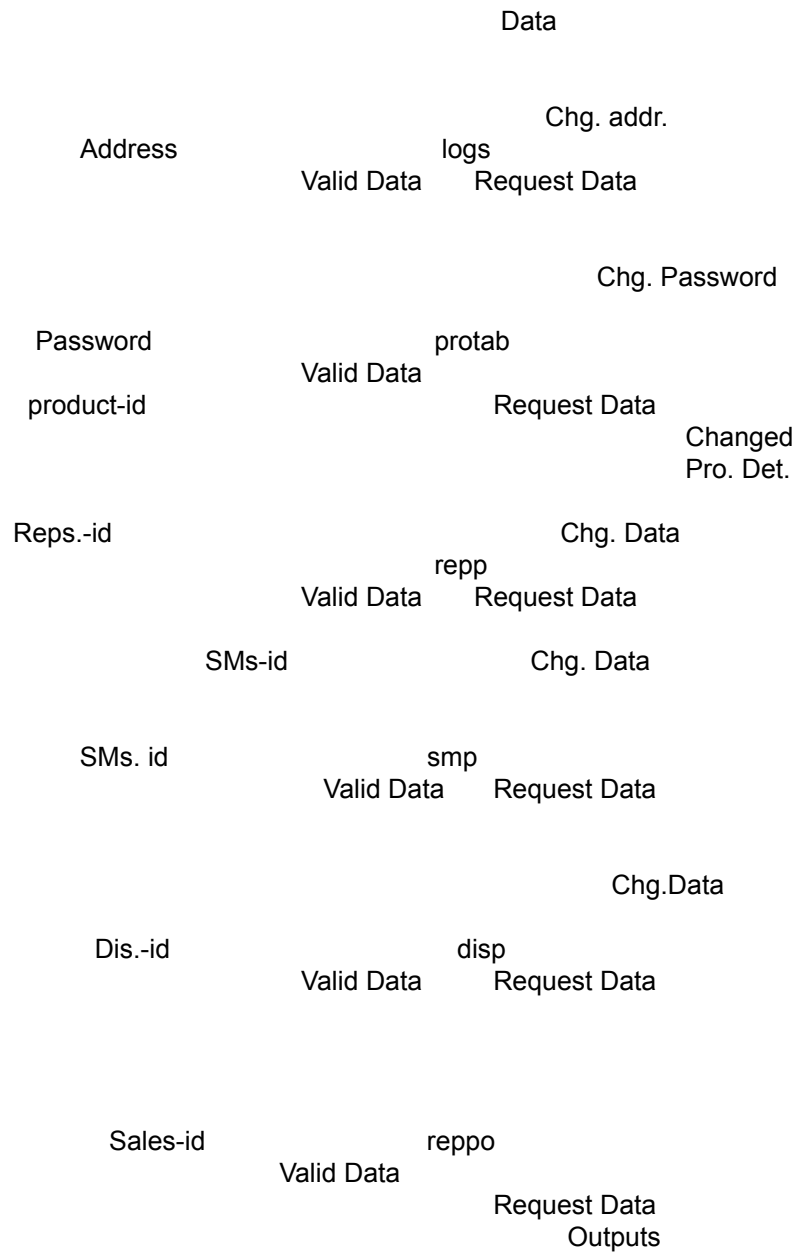
I' Level

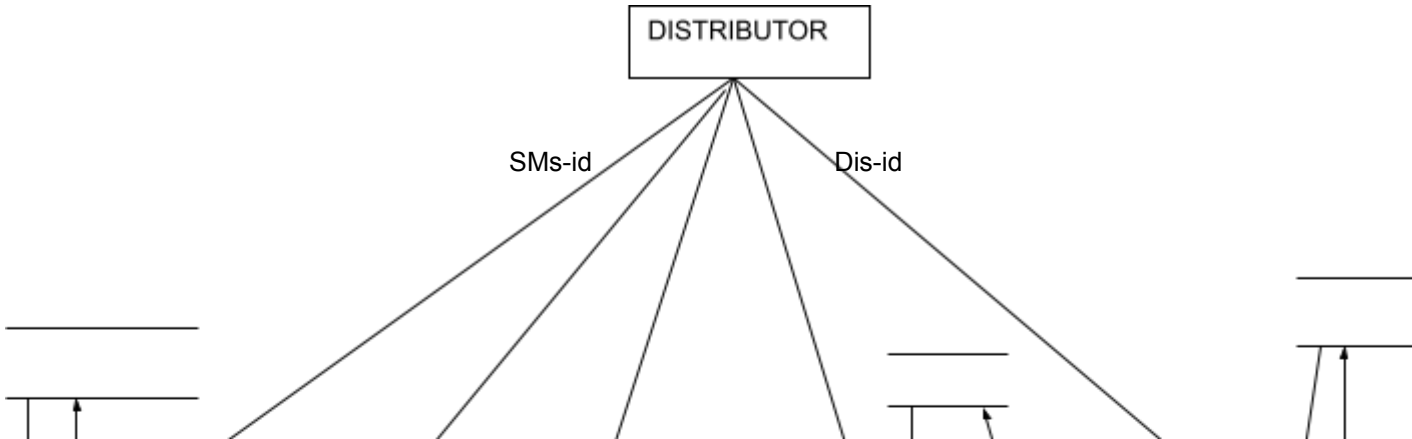
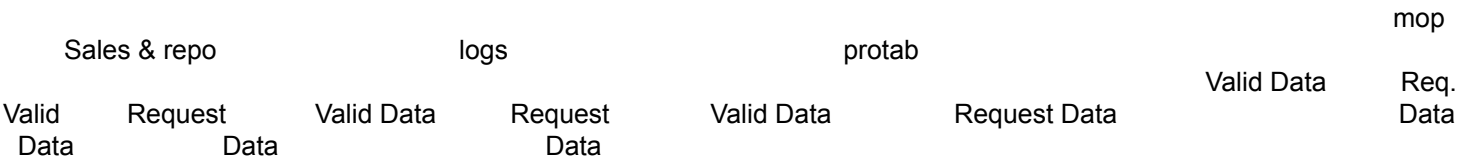


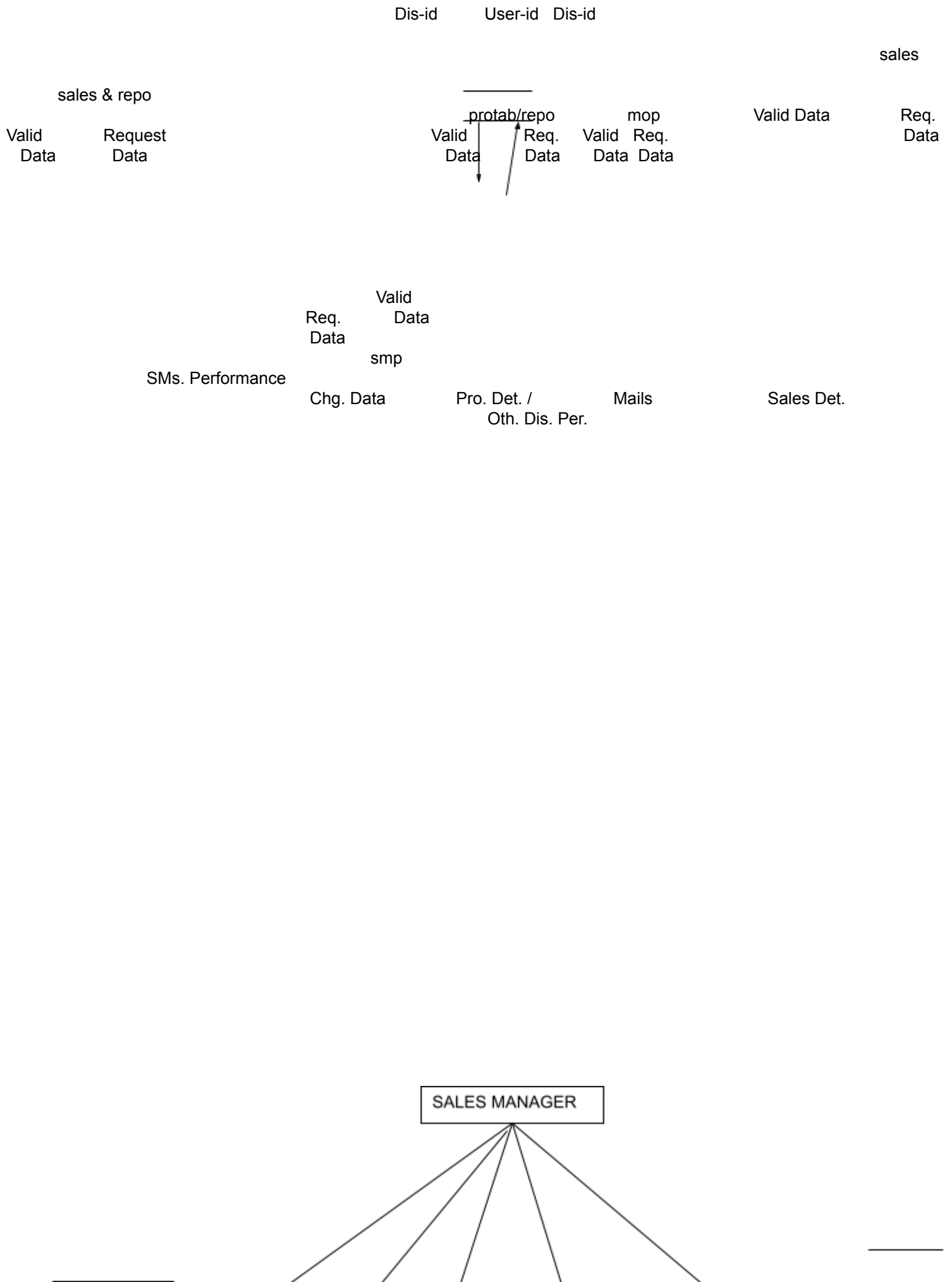


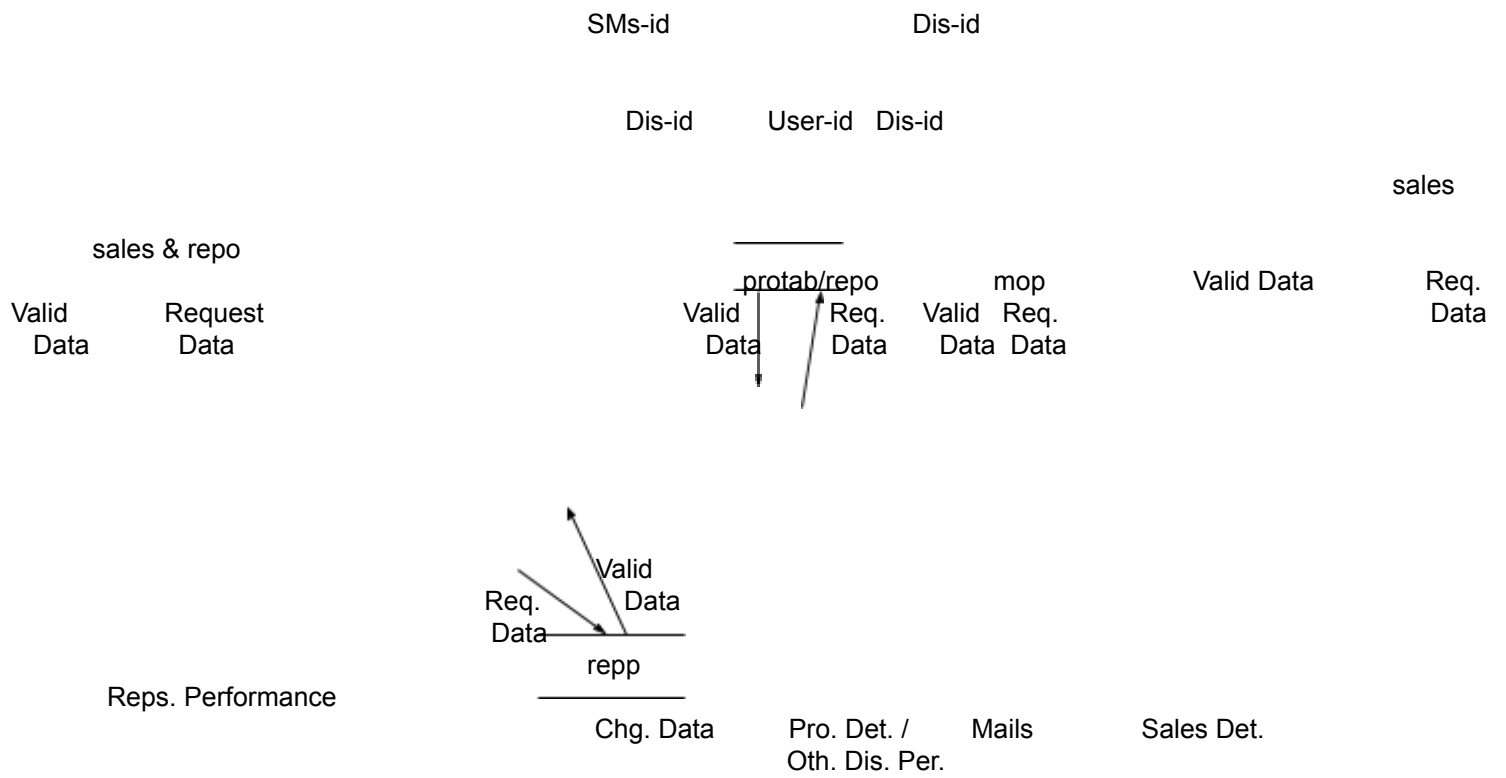
II' Level

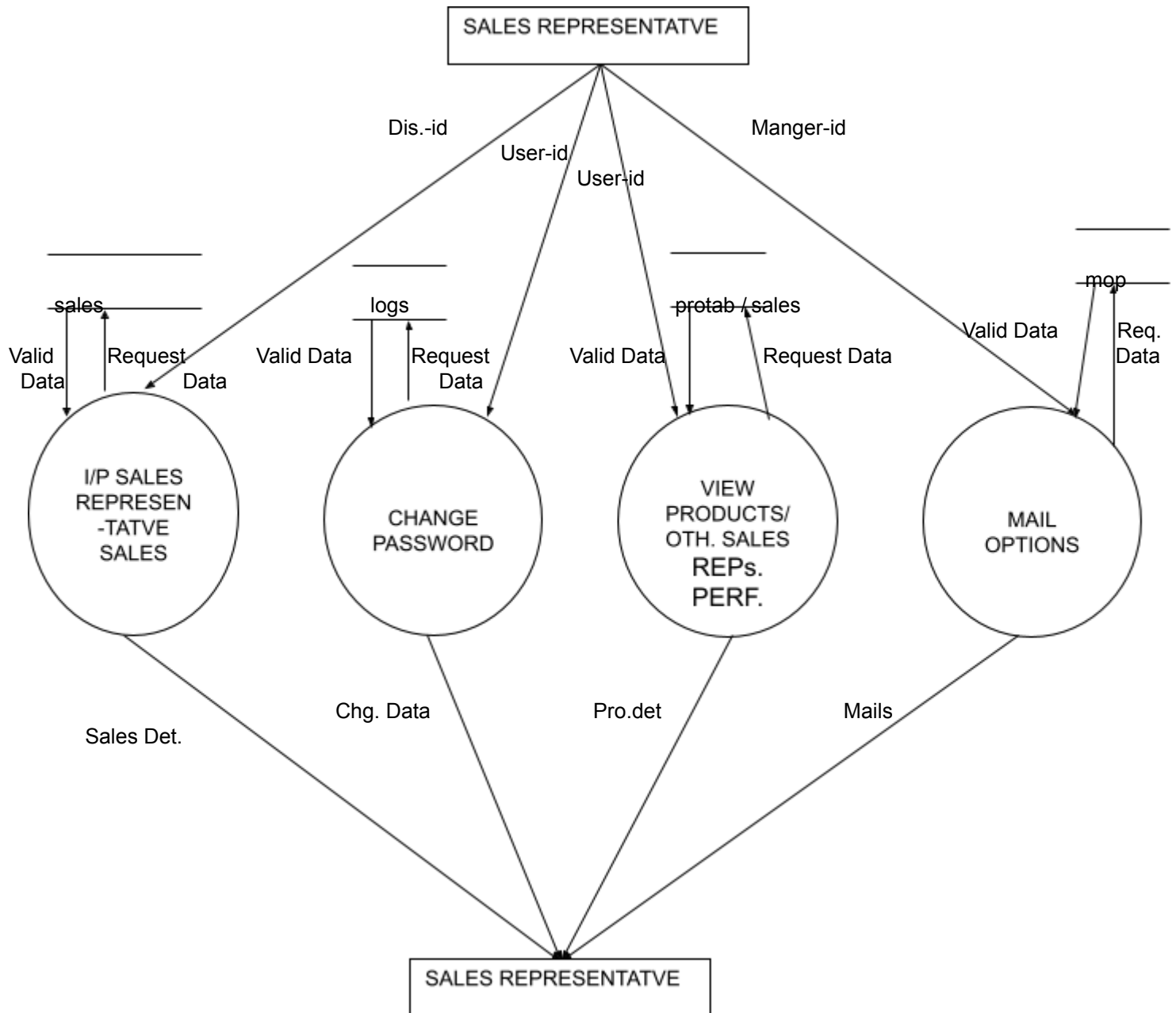












10. TESTING

Testing is a process of verifying whether the system developed satisfies the requirement specification given by the user. Testing is essential at each and every step of software development. The following are the important steps in testing

1. Construction of test data for the program
2. Analysis of results to detect program errors.
3. Localization of errors and modification of program to eliminate them i.e. debugging)

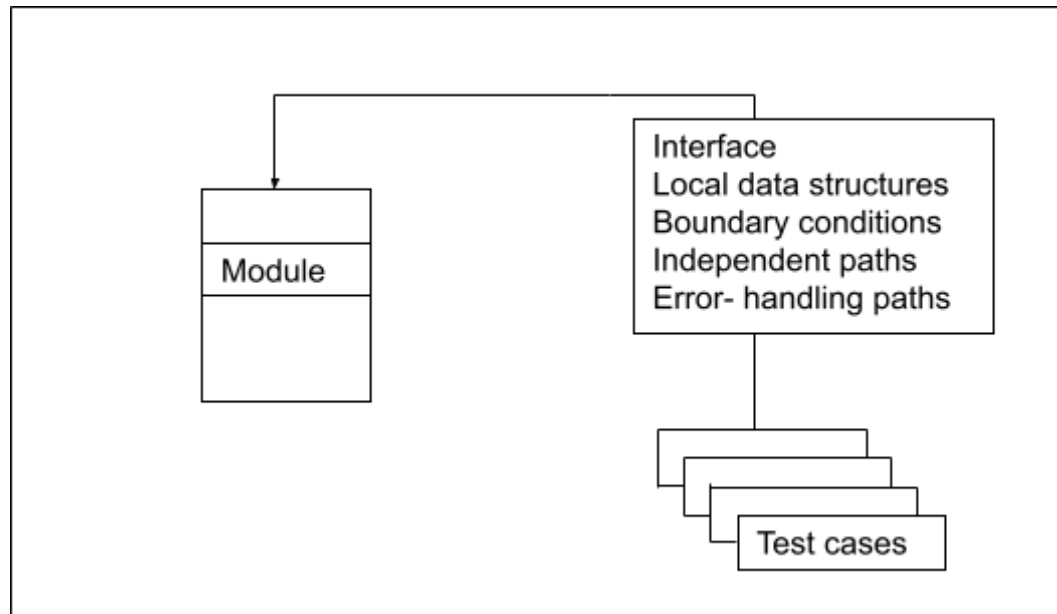
Types of Testing Done :

Unit Testing :

Unit testing focuses verification effort on the smallest unit of software design – the module. Using the design description as a guide, important control paths are tested to uncover errors within the boundary of the module . The relative complexity of tests and the errors detected as a result is limited by the constrained scope established for unit testing. The unit test is always white box –oriented, and the step can be conducted in parallel for multiple modules.

Unit Test Considerations

The tests that occur as part of unit testing are illustrated schematically in figure below. The module *interface* is tested to ensure that information properly flows and out of the program unit under tests. The local data *structure* is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that all modules operate properly at boundaries established to limit or restrict processing. All *independent* paths (basis paths) through the control structure are exercised to ensure that all statements in a module have been executed at least once. And finally, all *error – handling* paths are tested.



Tests of data flow across a module interface are required before any other test is initiated. If data do not enter and exit properly, all other tests are moot. In this text on software testing, Myers [MYE79] proposes a checklist for interface tests:

1. Numbering of input parameters equal to number of arguments?
2. Parameters and argument attributes match?
3. Parameter and arguments units' system match?
4. Number of arguments transmitted to called modules equal to number of parameters?
5. Attributes of arguments transmitted to called modules equal to attributes of parameters?
6. Units system of arguments to call modules equal to units systems of parameters.
7. Number attributes and order of arguments to built-in functions correct?
8. Any references to parameters not associated with current point of entry?

9. Input – only arguments altered?
10. Global variable definitions considerations across modules?
11. Constraints passed as arguments?

When a module performs external I/O, additional interface tests must be conducted.

Again, from Myers [MYE79].

1. File attributes correct?
2. OPEN/CLOSE statements correct?
3. Format specification matches I/O statements?
4. Buffer size matches record size?
5. Files opened before use?
6. End – of – line conditions handled?
7. I/O errors handled?
8. Any textual errors in output information?

The local data structure for a module is a common source of errors. Test cases should be designed to uncover errors in the following categories:

1. Improper or inconsistent typing
2. Erroneous initialization or default values
3. Incorrect (misspelled or truncated) values names
4. Inconsistent data types Underflow, overflow, and addressing exceptions.

In addition to local data structures, the impact of global data (e.g., FORTRAN COMMON) on a module should be ascertained (if possible) during unit testing.

Selective testing of execution paths is an essential task during the unit test. Test cases should be designed to uncover errors due to erroneous computations, incorrect comparisons, or improper control flow. Basis path and

loop testing are effective techniques for uncovering a broad array of path errors.

Among the more common errors in computations are (1) misunderstood or in correct arithmetic precedence, (2) mixed- mode operations, (3) incorrect initialization, (4) precision inaccuracy, and (5) incorrect symbolic representation of an expression. Comparison and control flow are closely coupled to one another (i.e., change of flow frequently occurs after a comparison). Test cases should uncover error such as (1) comparison of different data types, (2) incorrect logical operators or precedence, (3) expectation equality when precision error makes equality unlikely, (4) incorrect comparison or variables, (5) improper or nonexistent loop termination, (6) failure to exit when divergent iteration is encountered, and (7) improper modified loop variables.

Good design dictates that error conditions be anticipated and error handling paths set up reroute or cleanly terminate processing when an error occurs. Yourdon (YOU 75) calls this approach *antibugging*.

Unfortunately, there is a tendency to incorporate error handling into software and then never test it.

Among the potential error that should be tested when error handling is evaluated are:

1. Error description is unintelligible.
2. Error noted does not correspond to error handling.
3. Error condition causes system intervention prior to error handling.
4. Exception – condition processing is incorrect.
5. Error description does not provide enough information to assist in the location of the cause of the error.

Boundary testing is the last (and probably most important) task of the unit test step. Software often fails at its boundaries. That is, error often occurs

when the n th element an n -dimensional array is processed. When the i th repetition of a loop with l passes is invoked; when the maximum or minimum allowable value is encountered. Test cases that exercise data structure, control flow, and data values just below, at, and just above maxima and minima are very likely to uncover errors.

Integration Testing :

Integration testing involves Bottom-up integration, Top-down integration and sandwich integration strategy.

Bottom-up integrations the traditional strategy used to integrate the components of a software system into a functioning whole. Bottom – up integration consist of unit testing followed b subsystem testing, followed by testing the entire system. A subsystem consists of several modules that communicate with each other through well-defined interfaces. The primary goal of subsystem testing is to verify operations of the interfaces between modules in a subsystem.

Top-down integration starts with the main routine and one or two immediate subroutines in the system structure.

Sandwich integration is predominantly top-down, but bottom up techniques are used on some modules and subsystems. These alleviate many of the problems encountered in pure top-down integration at the subsystem and system level. In this system sandwich integration testing is carried out for its validation of the program units.

Acceptance Testing :

Acceptance testing involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements.

Functional test cases involve exercising the code with nominal input values for which expected results are known. It is tested by giving different input values.

Performance testing determines the amount of executing time spent in various paths of the program unit, program throughput, the response time and device utilization by the program unit. With respect to the system performance testing is based on the maximum volume of existing data which the system can handle with an effective throughput and efficient utilization of the system resources.

Software system is developed in the above manner is one that satisfied the user needs, confirms to its requirements and design specifications, and exhibits an absence of errors.

User Interface Testing :

At the culmination of integration testing, software is completely assembled as a package, interfacing errors have been uncovered and corrected, and a final series of software tests – Validation testing – may begin. Validation succeeds when the software functions in a manner that can reasonably expected by the customer.

Validation Test Criteria :

Software validation is achieved through a series of black box tests that demonstrate conformity with requirements. A test plan outline the classes of test to be conducted and a test procedure defines specific test will be used to demonstrate conformity with requirements.

After each validation test case been conducted, one of two possible conditions exist (1). The function or performance characteristics confirm to specification and or expected or (2) A deviation from specification is and

covered and deficiency list is created. Deviation or error discovered at this stage in a project can rarely be corrected prior to scheduled completion. It is often necessary to negotiate with the customer to establish a method for resolving deficiencies.

ERROR MESSAGES

Error Messages and warnings are “bad News” delivered to the user’s iterative systems when something has gone away or wrong. Therefore in this developed software there are some messages which will be displaying while using this developed software if the user goes wrong. These messages are used to help the use for the better use.

When the user enters the wrong password the displayed message would be “Invalid Password ! Please try again. “. While adding information for various modules no field should be empty when saving the information in the database otherwise corresponding error messages are given immediately. While updating the information if any field is left empty then the messages are displayed accordingly like “Select the Task Title”, “Please Enter Numeric Values” etc.

Thus these error messages are displayed in simple language in order to understanding of the user not to increase the frustration.

Like this for all the modules in the project has been tested by giving different input values and which has been compared with the expected results with success.

In this project the various types of testing are conducted as follows.

Performance Testing

In this testing, the amount of executing time is very less because, the coding has been done in a java servlets and HTML has been used for the

front end designing. So that the execution time is less and the hit rate will be faster.

Validation testing

The validation testing has been given wherever it is necessary and the appropriate error message has been given. The example part is given below for both the Client-side validation and Server-side validation.

Index	Test case description	Registered Input	Expected Result	Actual Result
Test1	Checking validity User Id and password from the user.	Giving a valid User Id and password	Successful login can enter into the module.	User-id & password are checked with logs table.
Test2	Login Page.	User gives a Invalid userId.	Concerned error message.	User alerted with the message "Invalid UserId".
Test3	Login Page.	User gives a Invalid password.	Concerned error message.	User alerted with the message "Invalid password"
Test4	Login Page	User do not enters a Username	Concerned error message.	User alerted with the message "Please enter a UserNane".

Test5	Checking the access rights for modules	Giving a valid User Name and password	Successful login and display the modules that the user has access rights	User-id & password are checked with logs table.
Test6	Checking no field is left empty before saving the details that the user enter for different modules	Giving a valid details in the form	Successful storage of details in the respective tables	The details that are entered are validated and stored in different tables.
Test 7	Checking in the modify page,so that the values are displayed in the respective fields.	Giving only the valid details in the form	Successful updation of details in the respective tables.	The details that are modified are updated in the respective tables.

11. IMPLEMENTATION PHASE

After testing the system and find it successful to put the new system into operation. There are different techniques that can be used to replace an existing system with the new system.

- **Direct Change Over**

In this technique the existing system is replaced by the proposed system after ensuring that the system objectives are met. For this application this

method is adopted. This software is successfully demonstrated to HSL organization.

- **Parallel run**

The Proposed system is put into operation in parallel to monitor the performance. The system has produced the expected results.

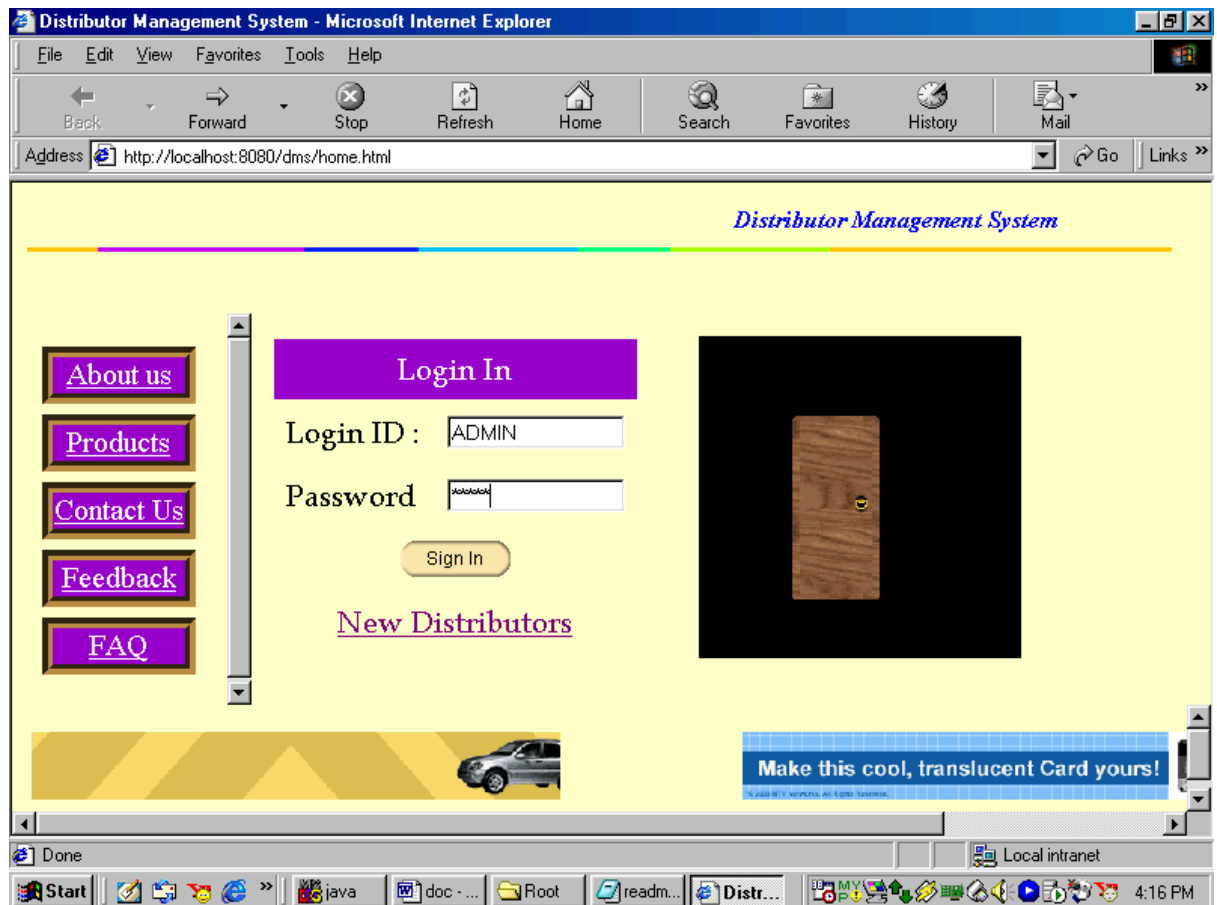
- **Pilot**

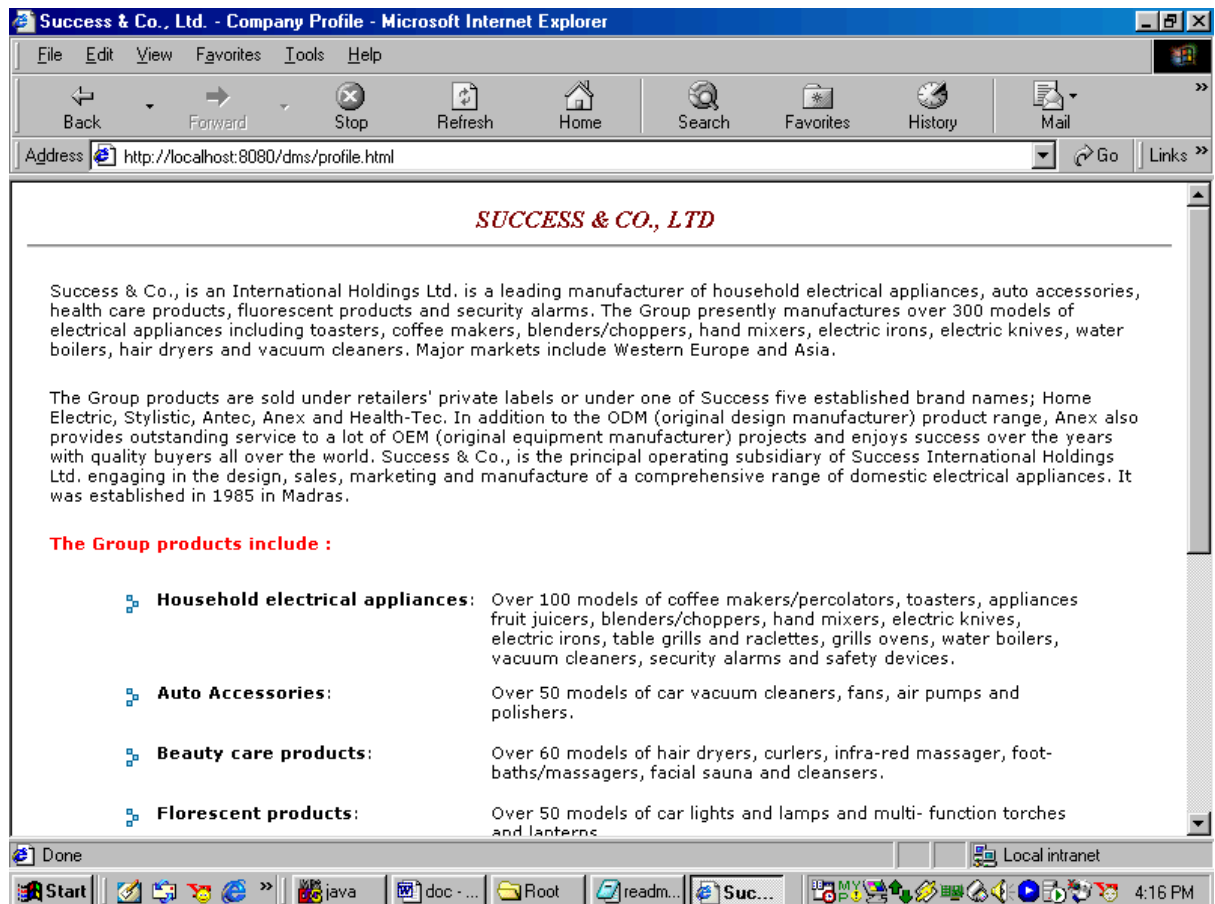
In pilot run, the system is tested with available results. The performance of the system is studied with the latest data.

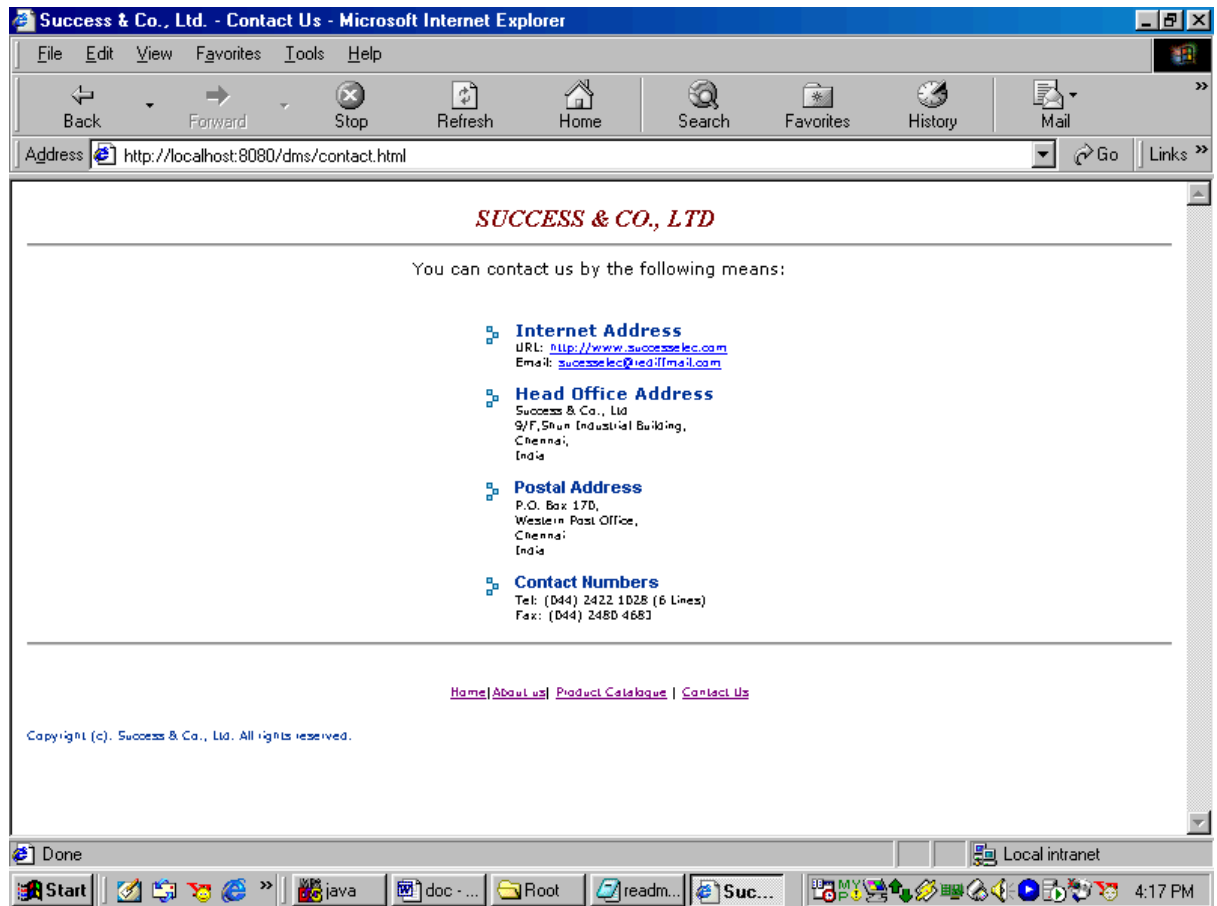
- **Staged change over**

In this technique, the proposed system is implemented in several stages. The existing system becomes unoperational if all stages are successfully implemented.

SCREEN LAYOUTS









Success & Co., Ltd. - Contact Us - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address  http://localhost:8080/dms/feedback.html Go Links

SUCCESS & CO., LTD

 **Feedback Form**

Name *:

Contact No *:

Fax No:

Email:

Subject:

Enquiry *:

Submit Clear

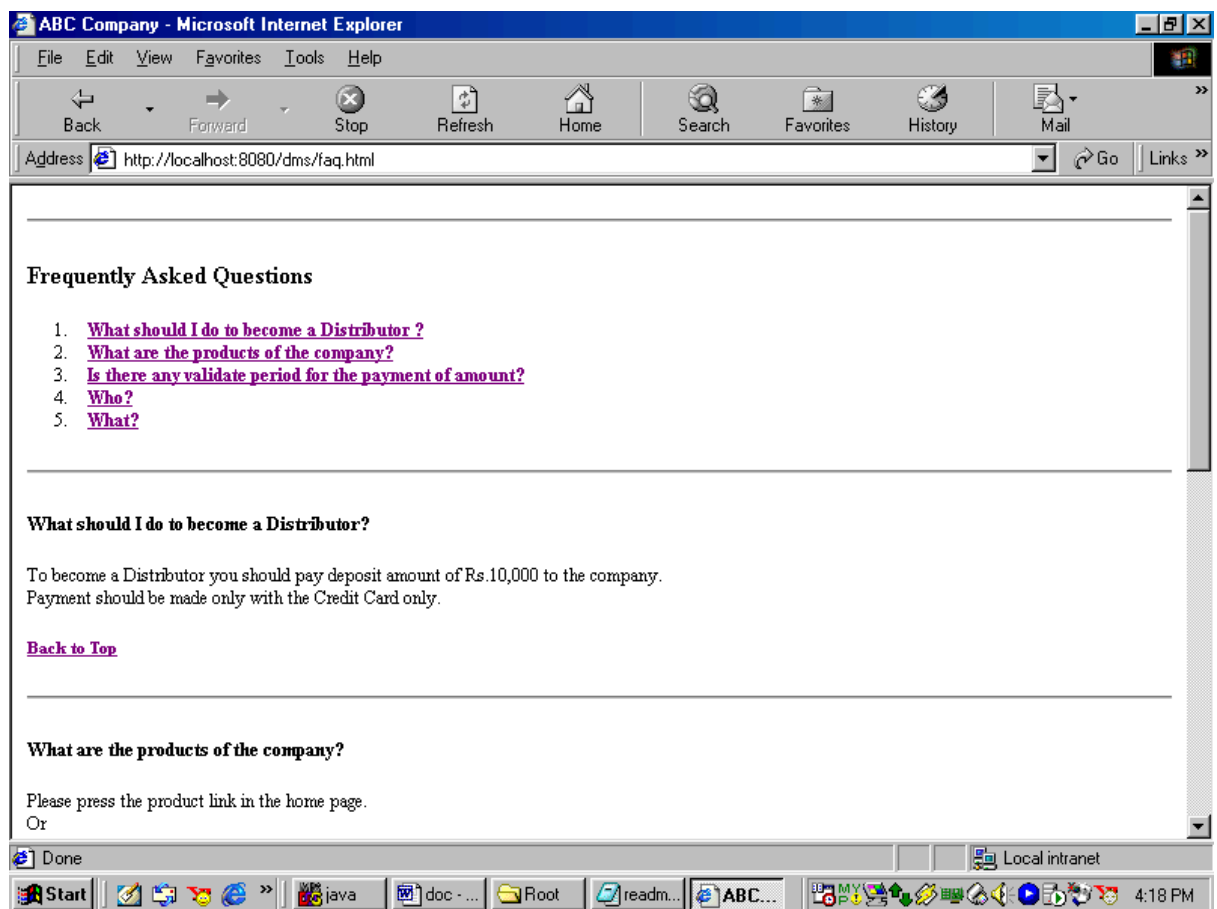
*: required

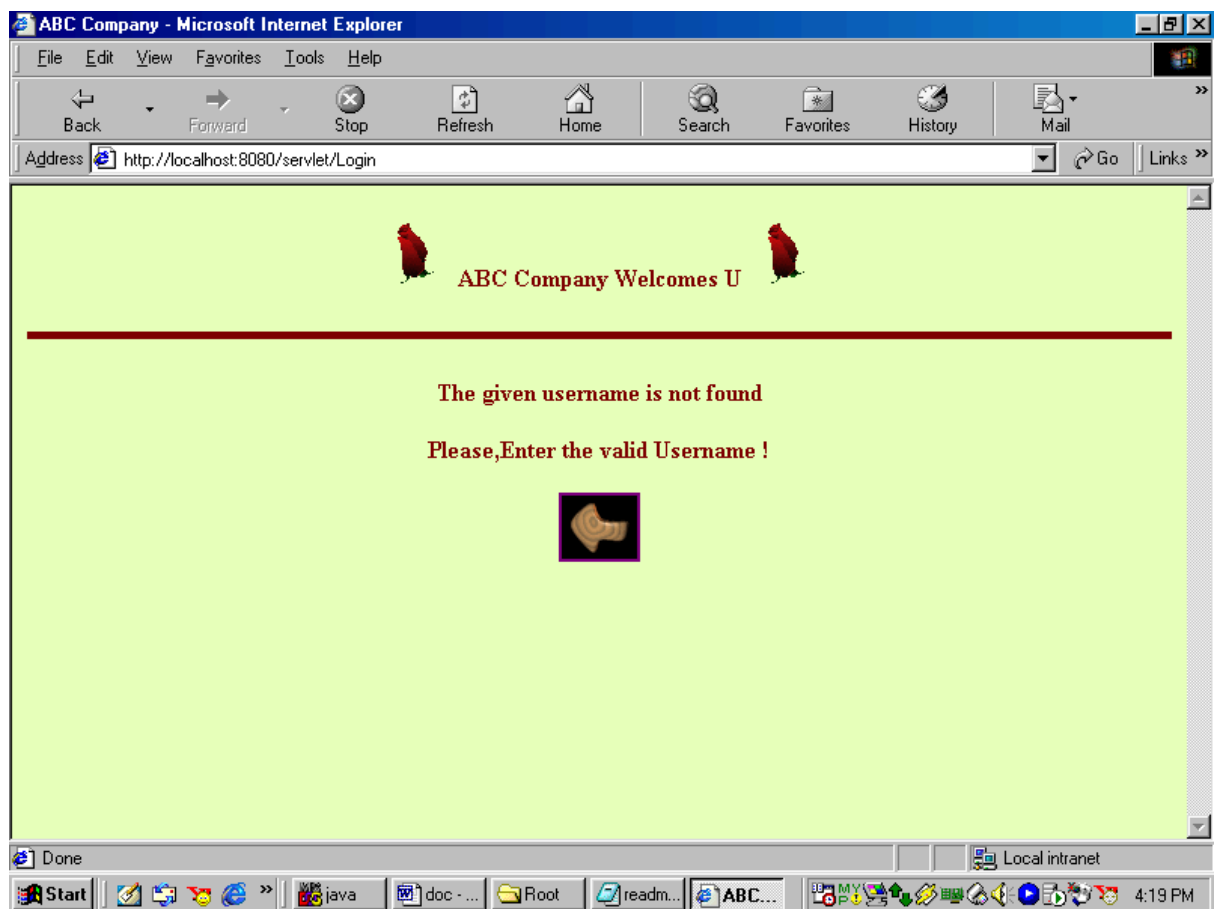
[Home](#) | [About us](#) | [Product Catalogue](#) | [Contact Us](#)

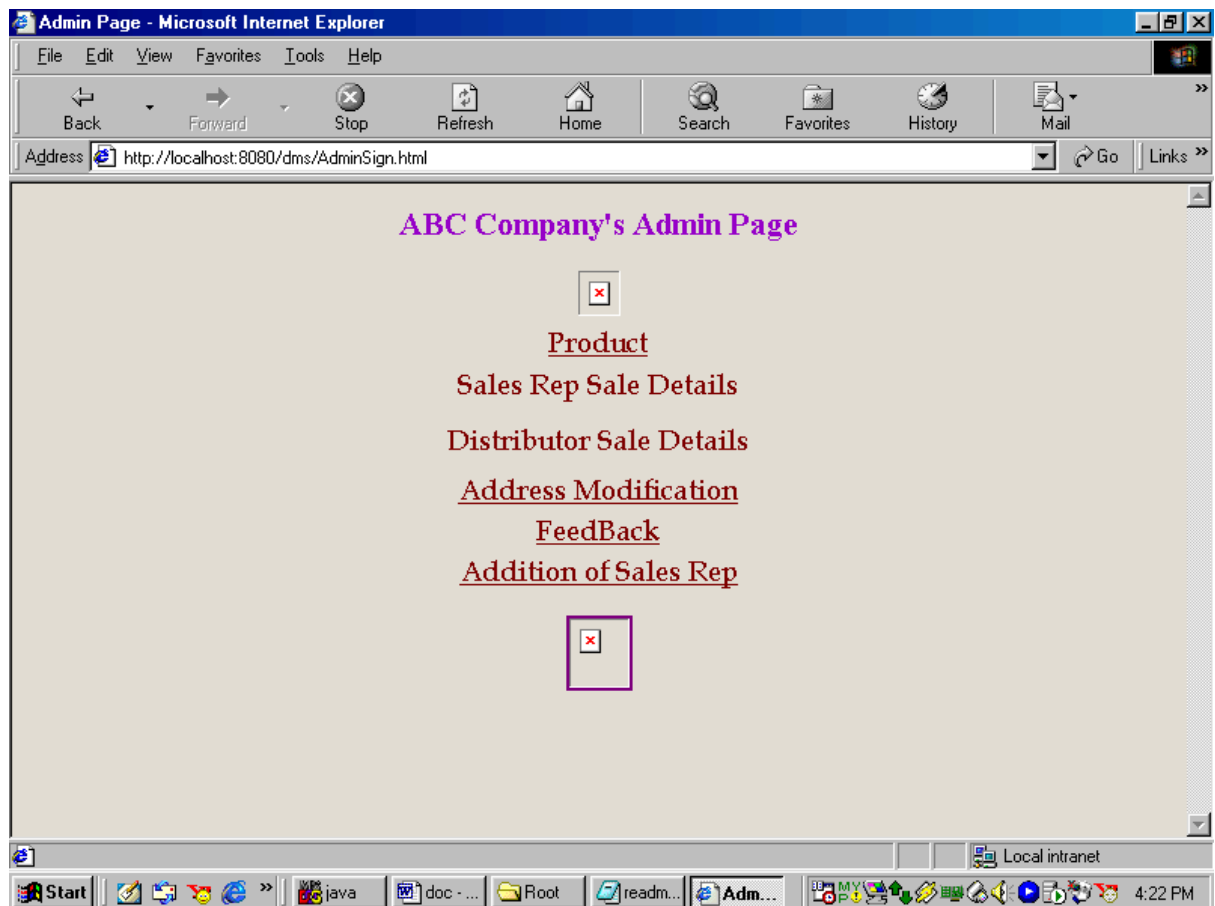
Copyright (c). Success & Co., Ltd. All rights reserved.

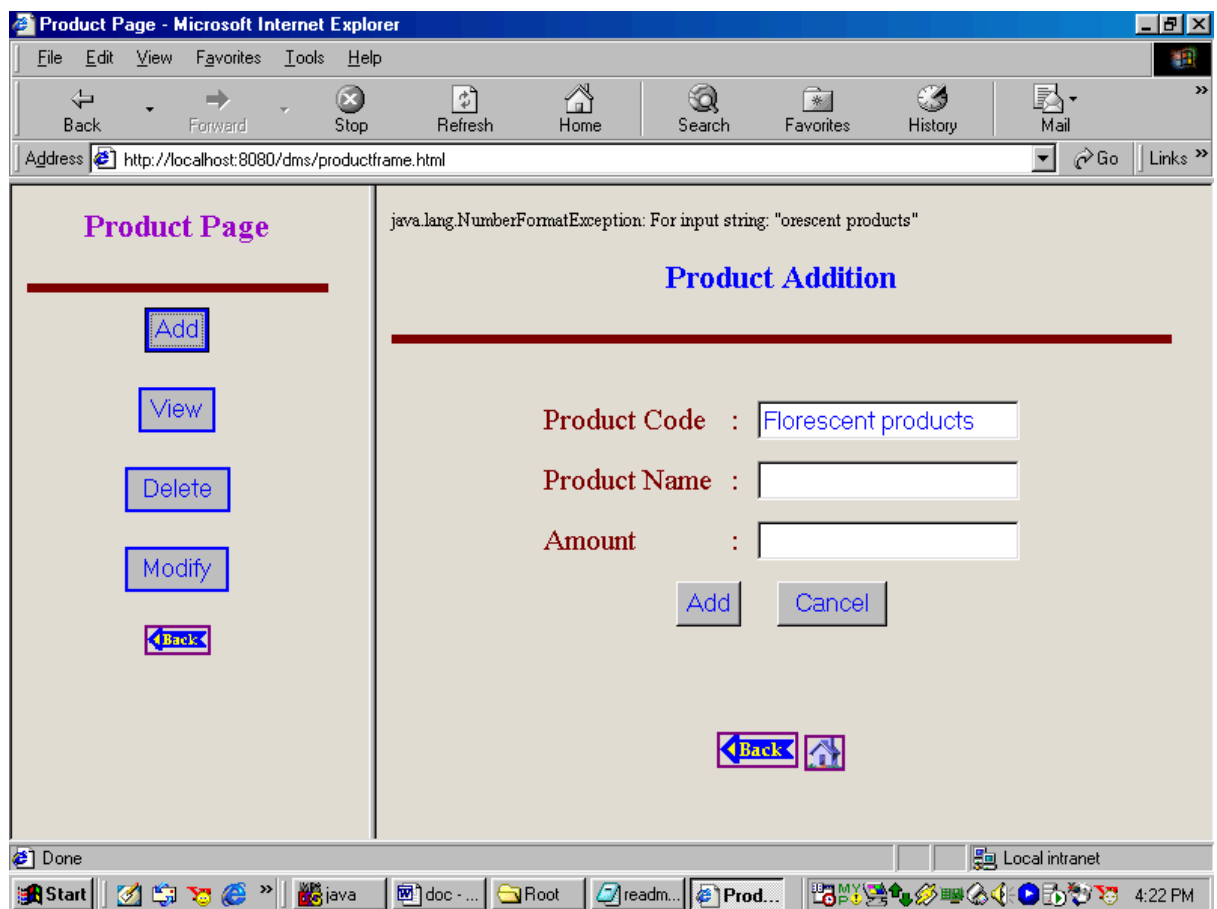
Done Local intranet

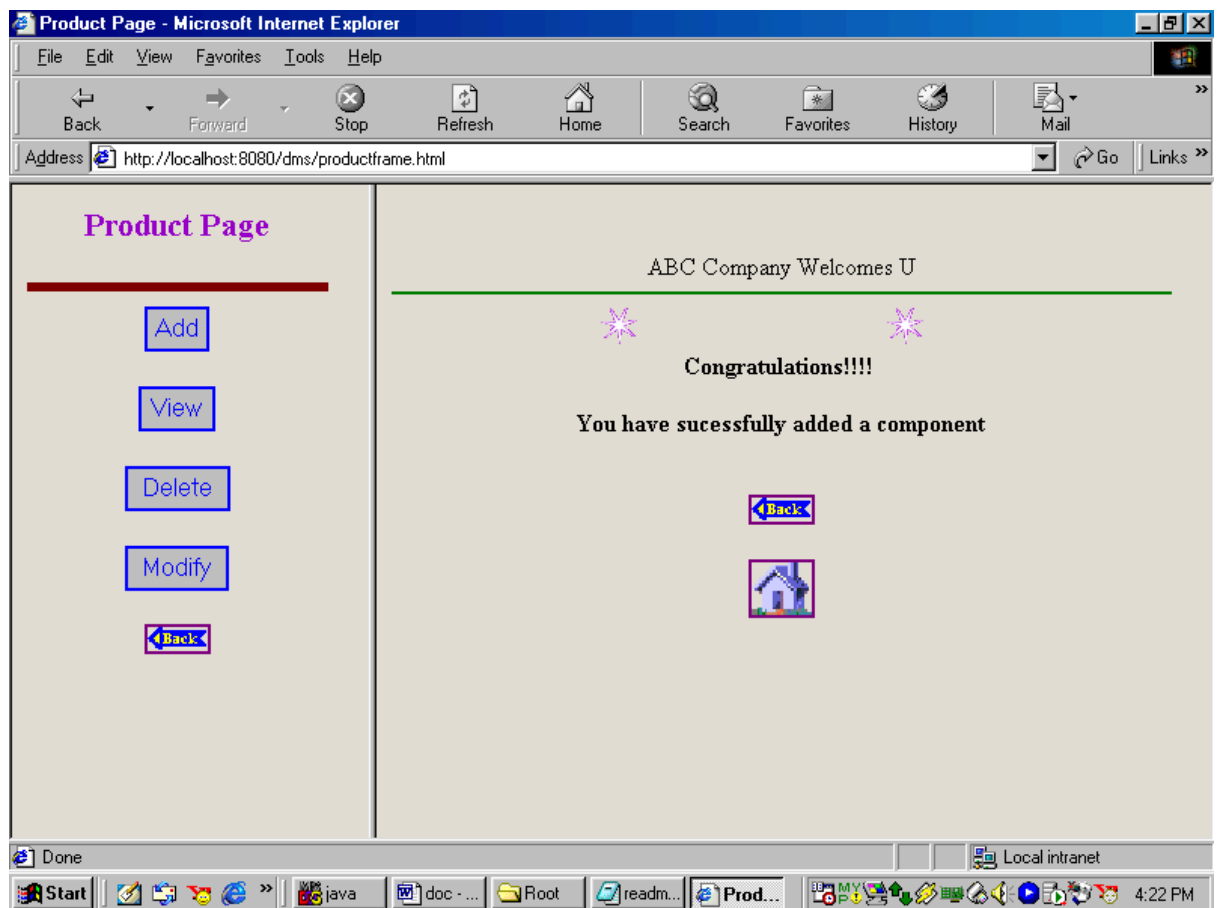
Start java doc - ... Root readm... Suc... 4:18 PM











Product Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address http://localhost:8080/dms/productframe.html Go Links

Product Page

Add

View

Delete

Modify

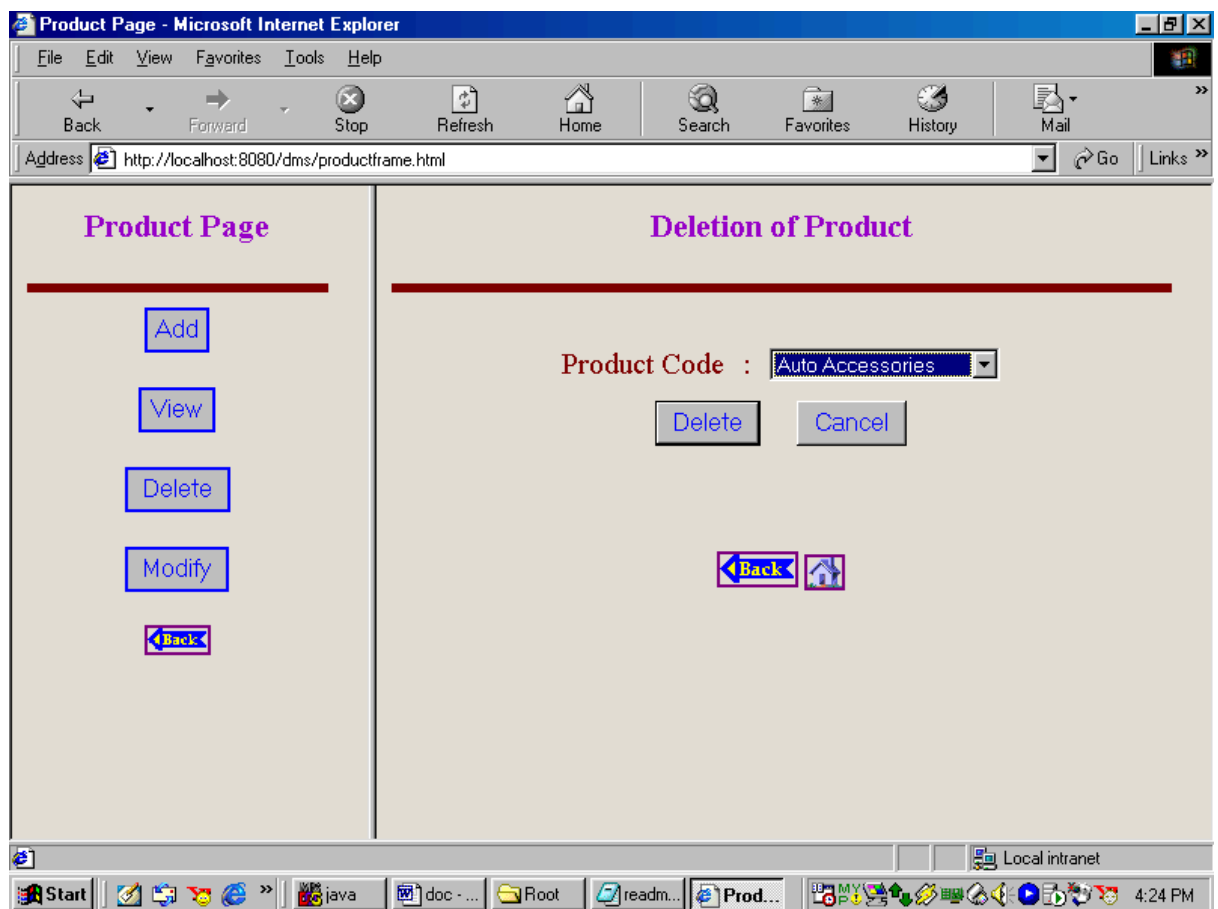
Back

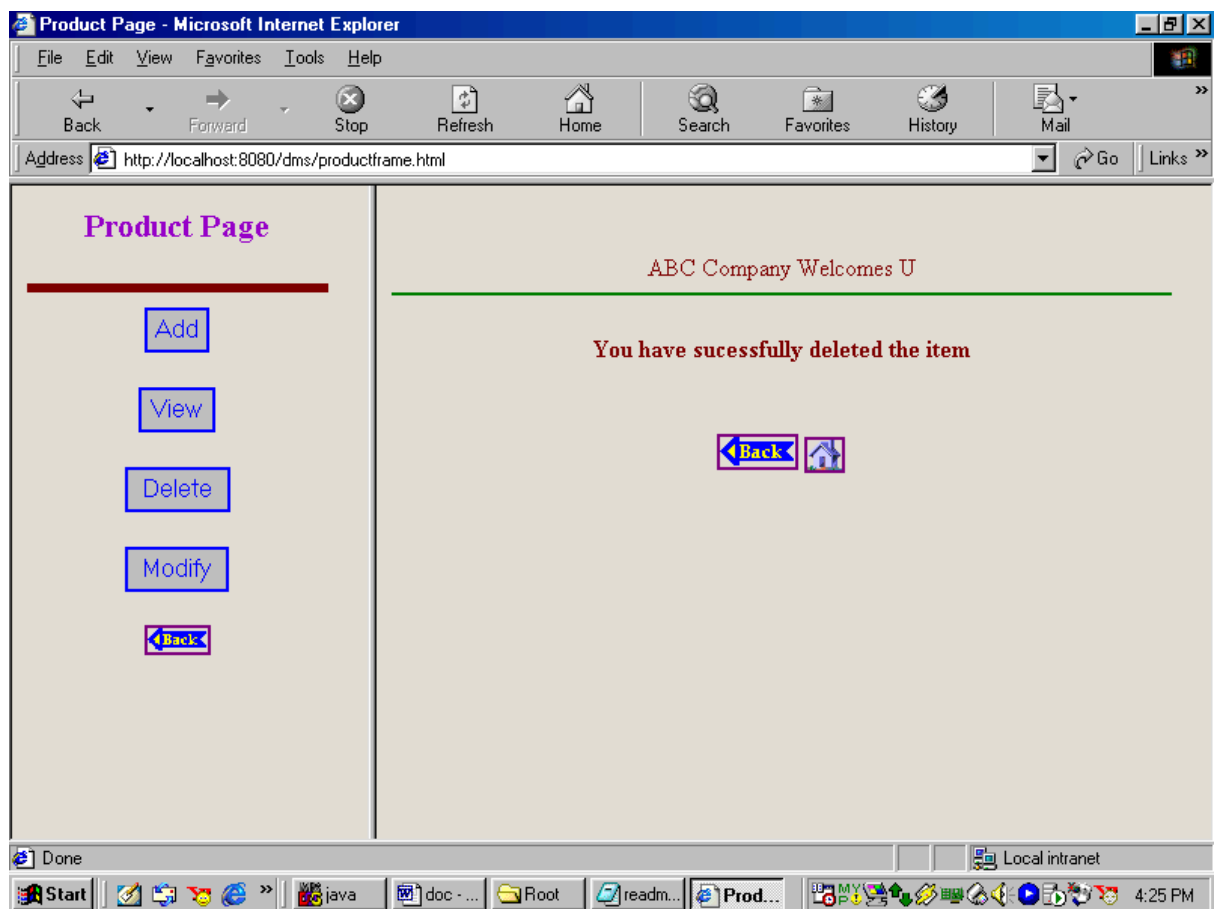
Product View

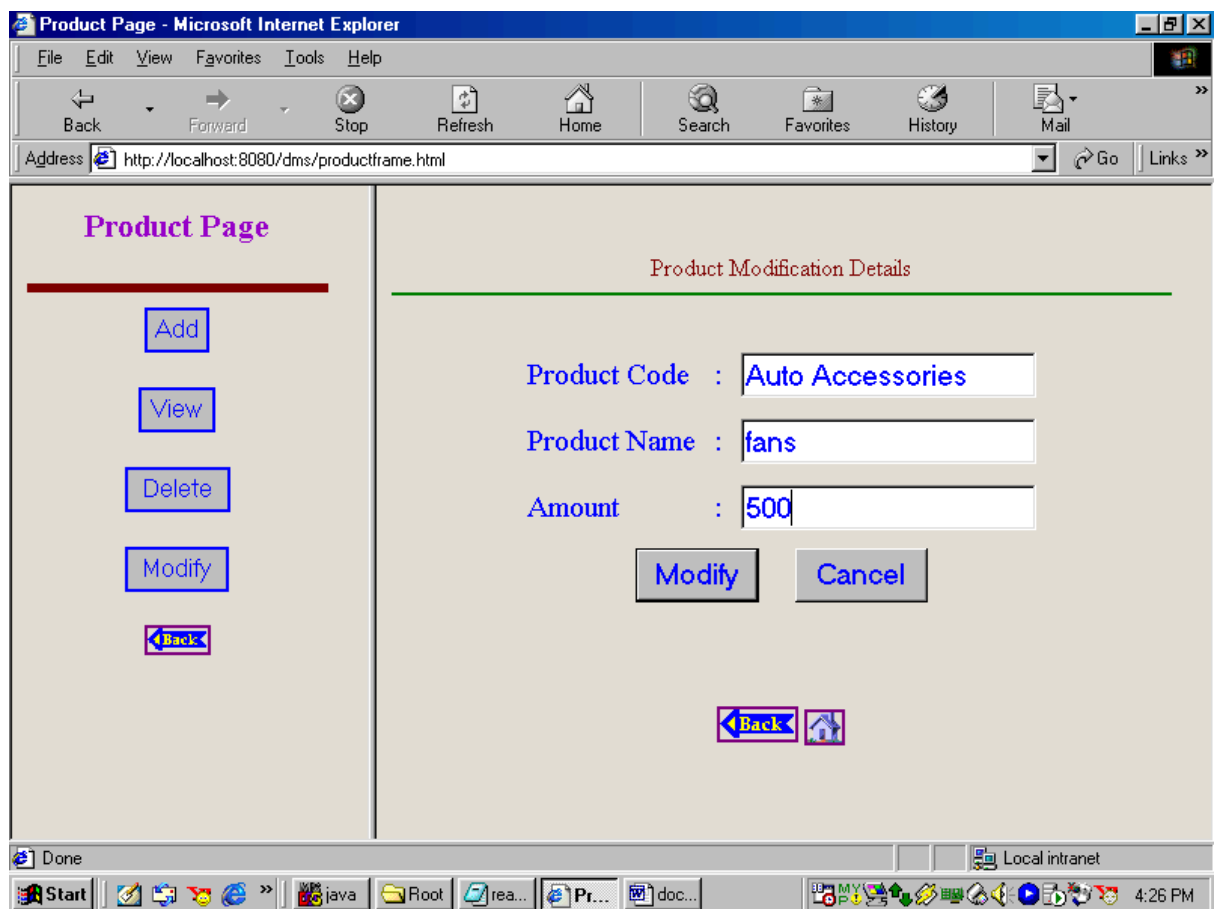
Product Code	Product Name	Amount
Auto Accessories	vacuum cleaners	800
Auto Accessories	fans	1400
Auto Accessories	air pumps	300
Auto Accessories	polishers	200
Beauty care products	hair dryers	150
Beauty care products	curlers	200
Beauty care products	infra-red massager	255
Beauty care products	massagers	335
Beauty care products	facial sauna	222
Florescent products	car lights	460
Florescent products	car lamps	200
Florescent products	torches	100

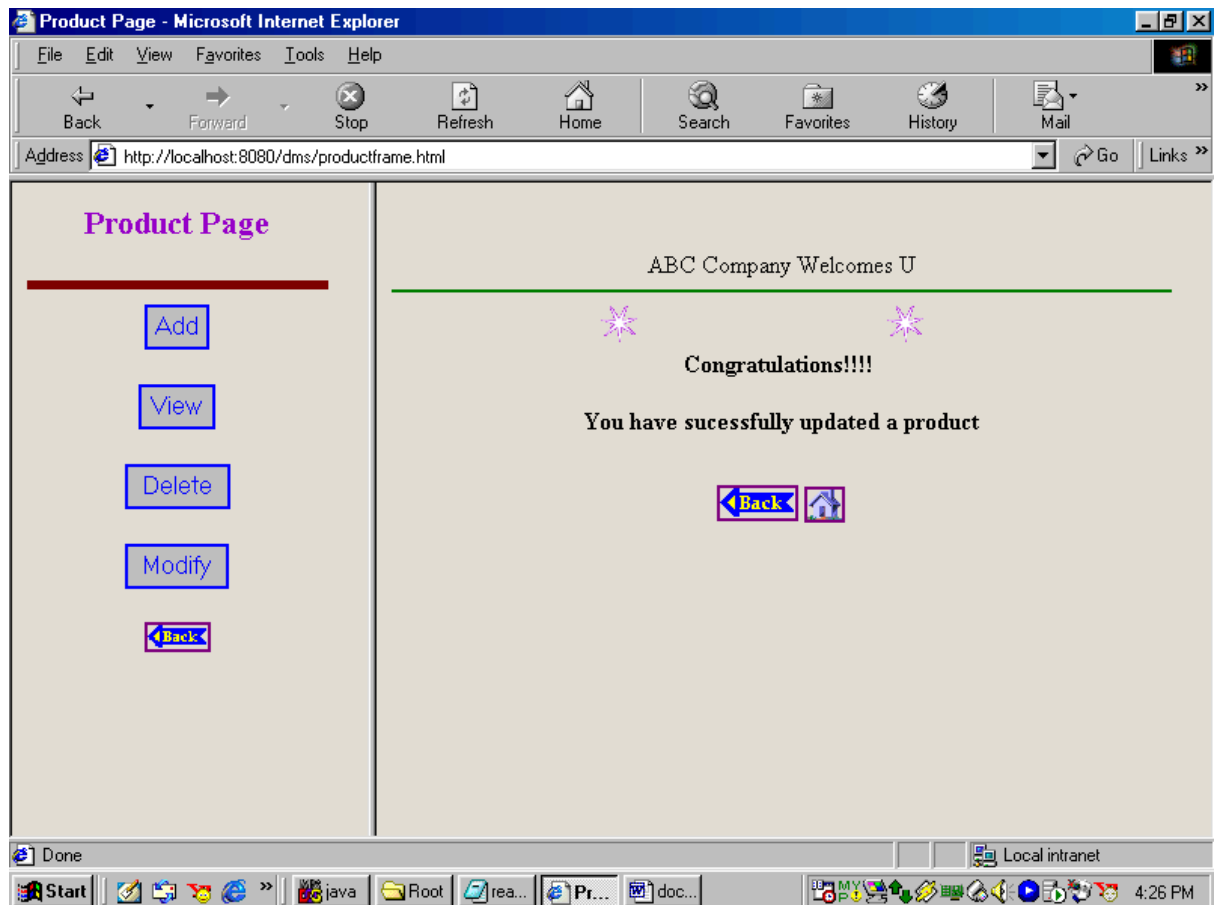
Done Local intranet

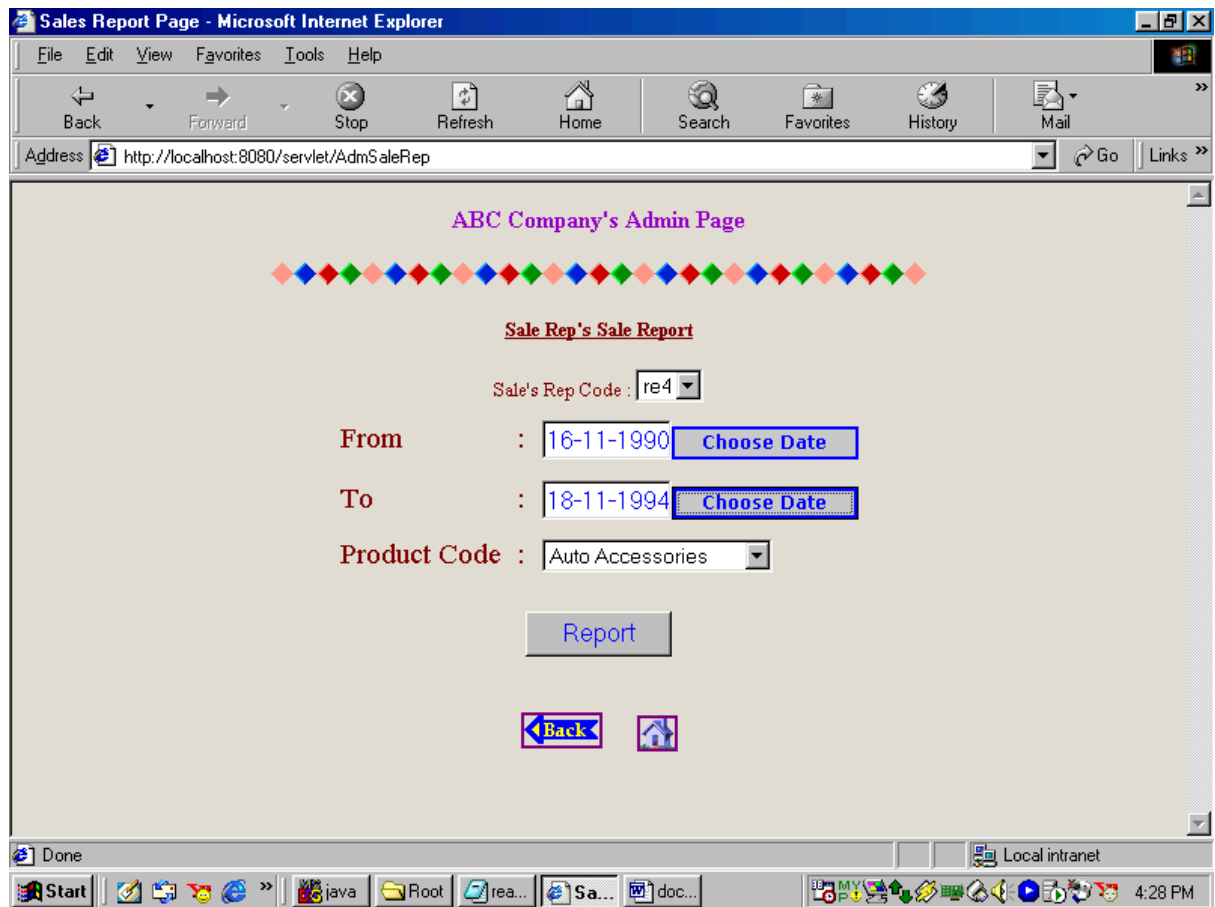
Start java doc - ... Root readm... Prod... 4:23 PM

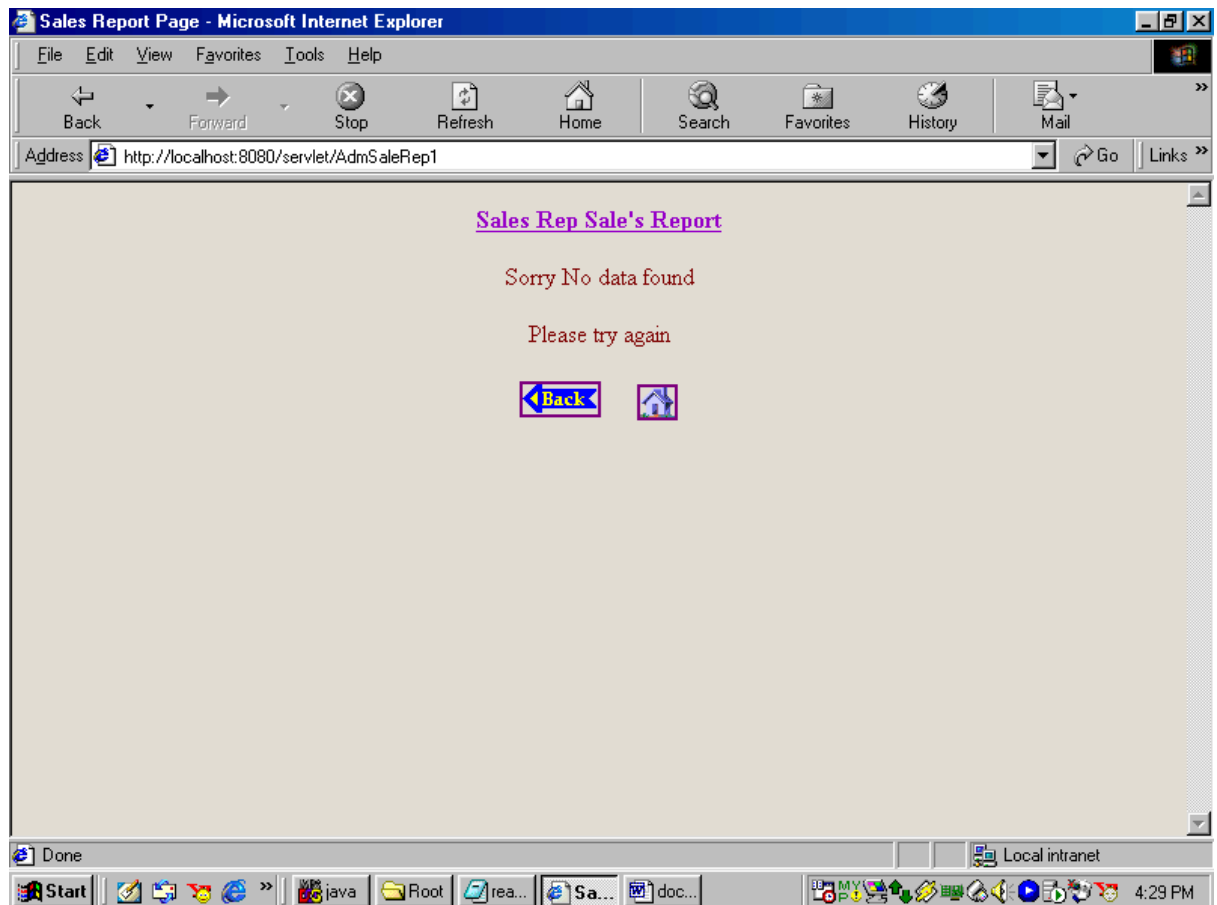


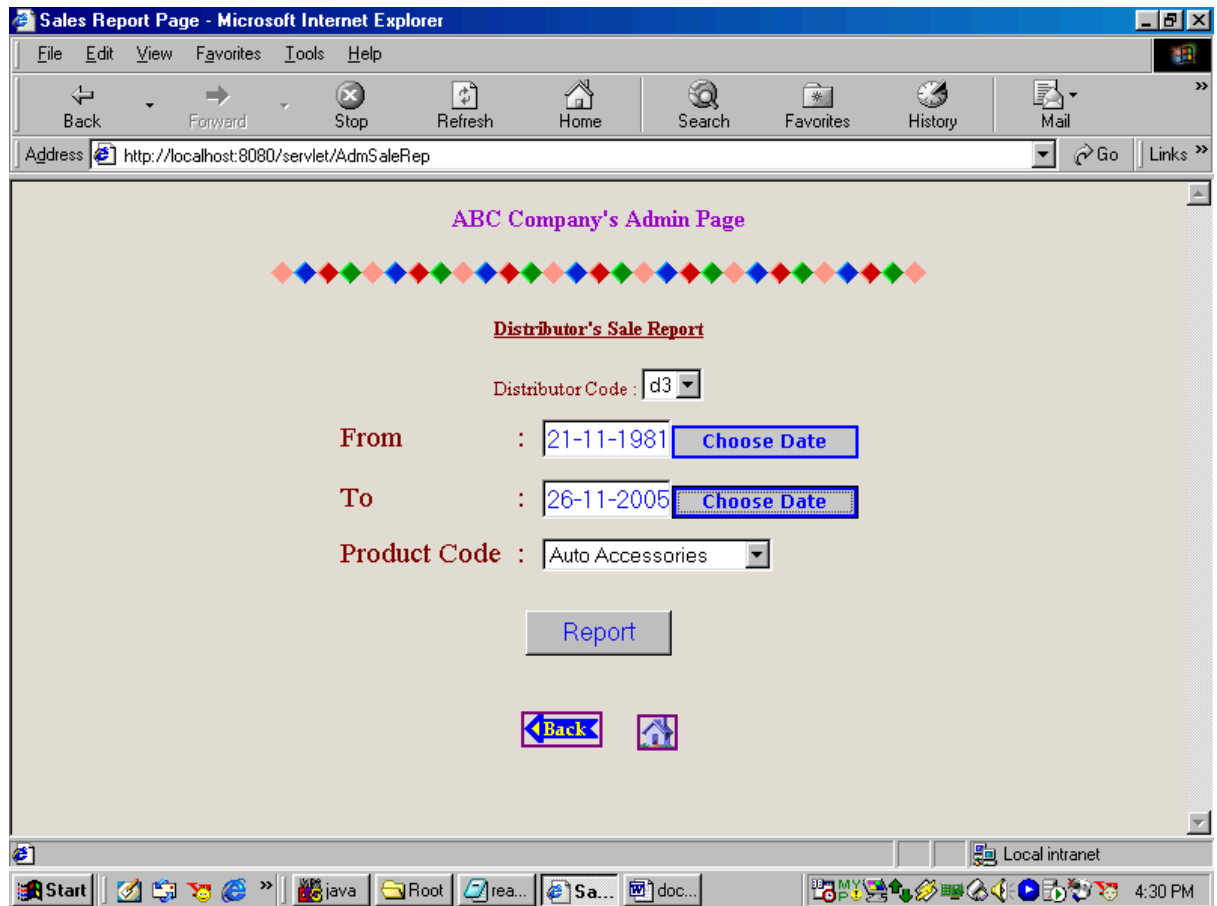


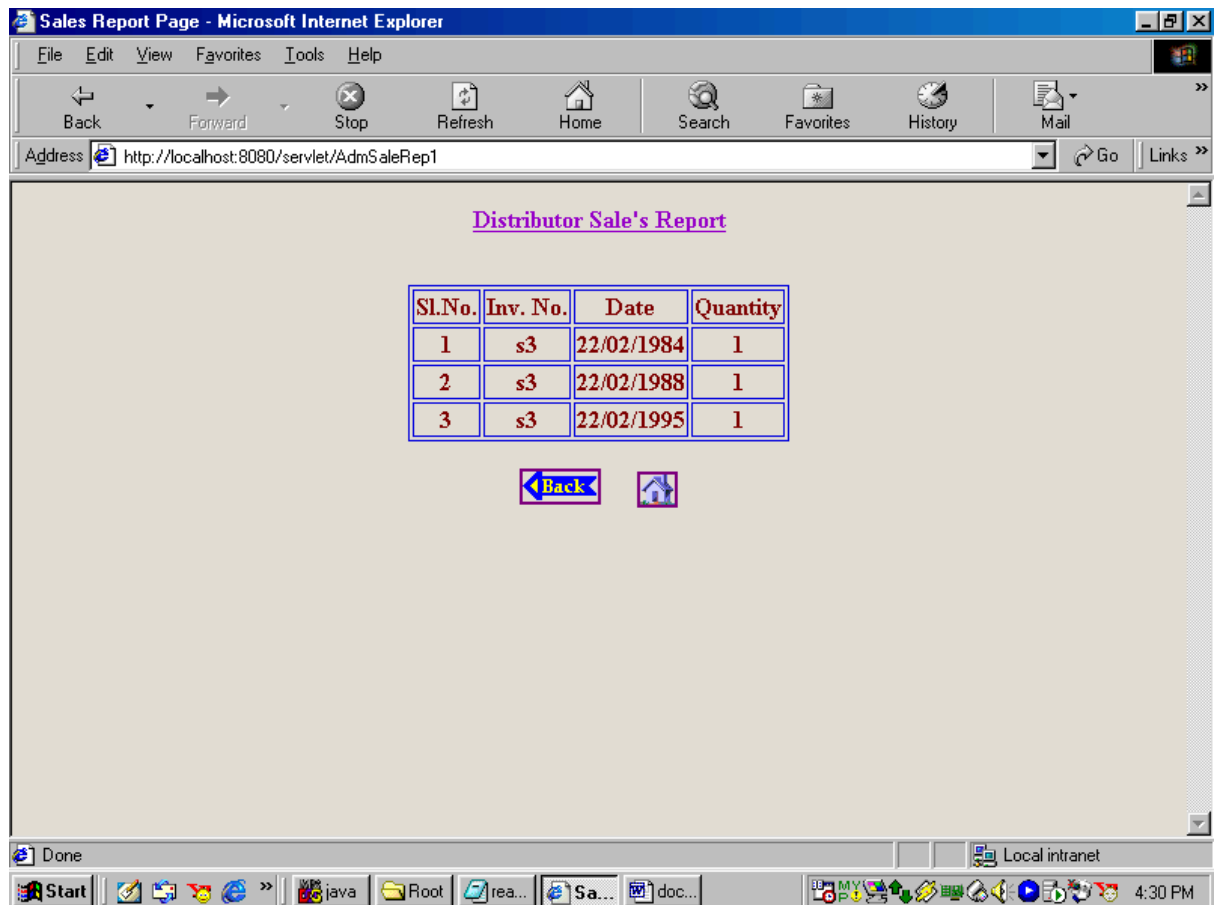


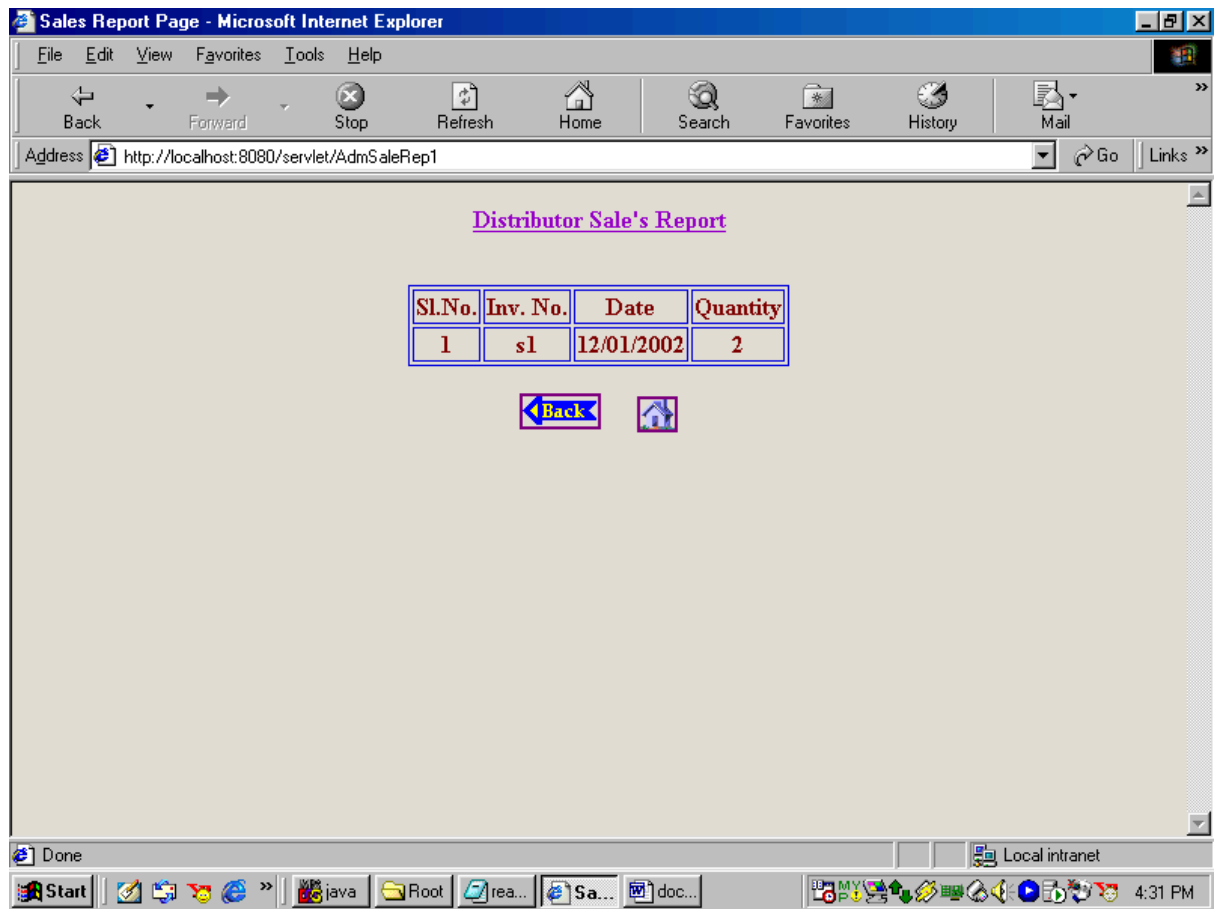


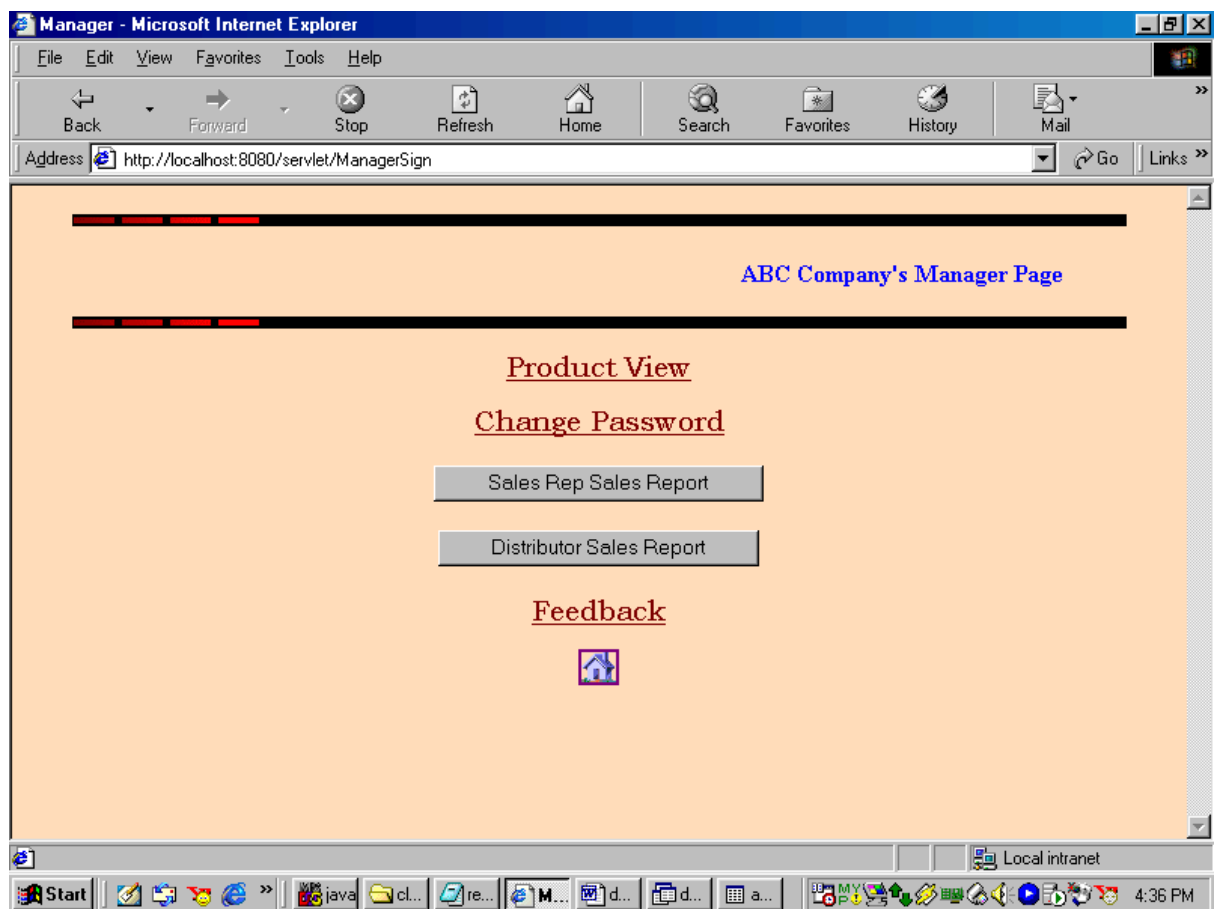


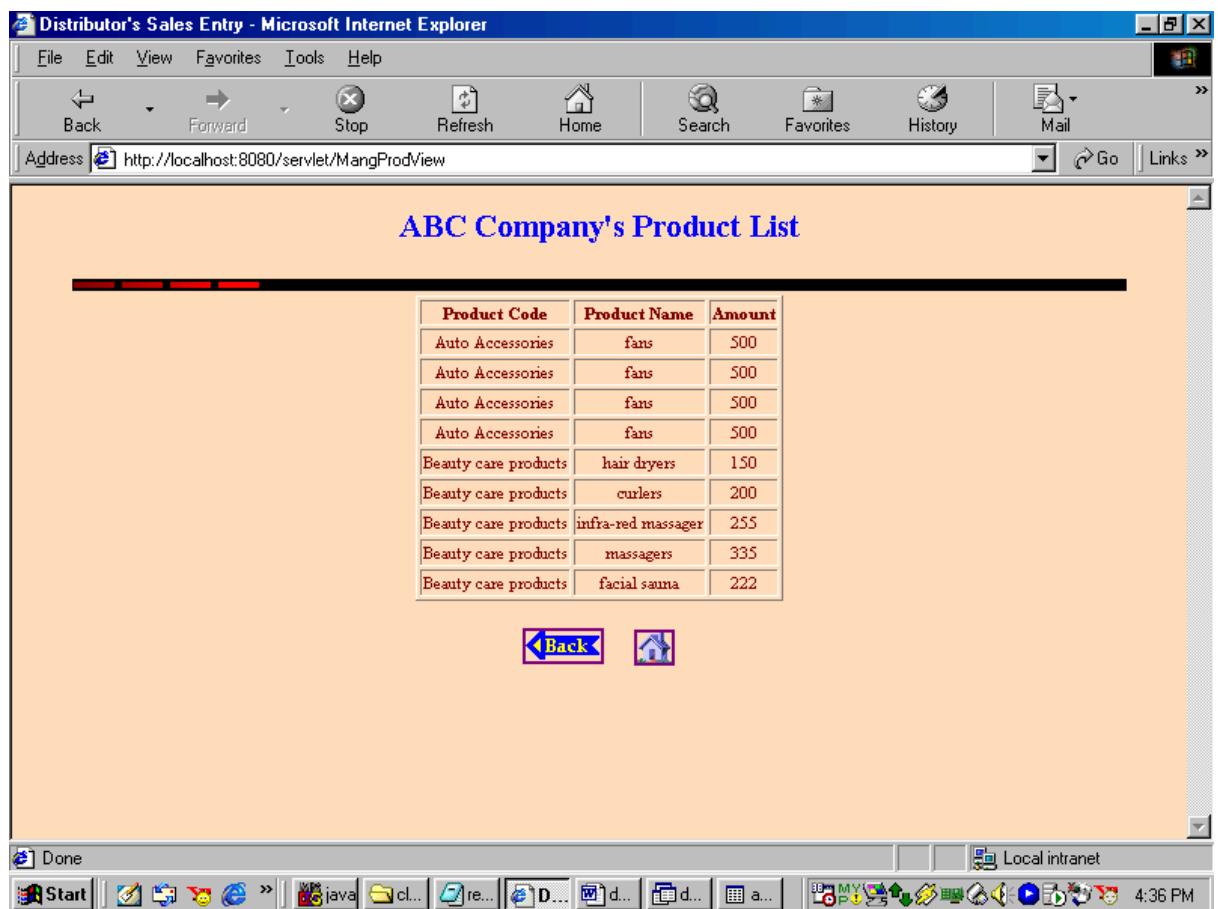












Password change - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/dms/passchange.html> Go Links

ABC Company

Password Change Page

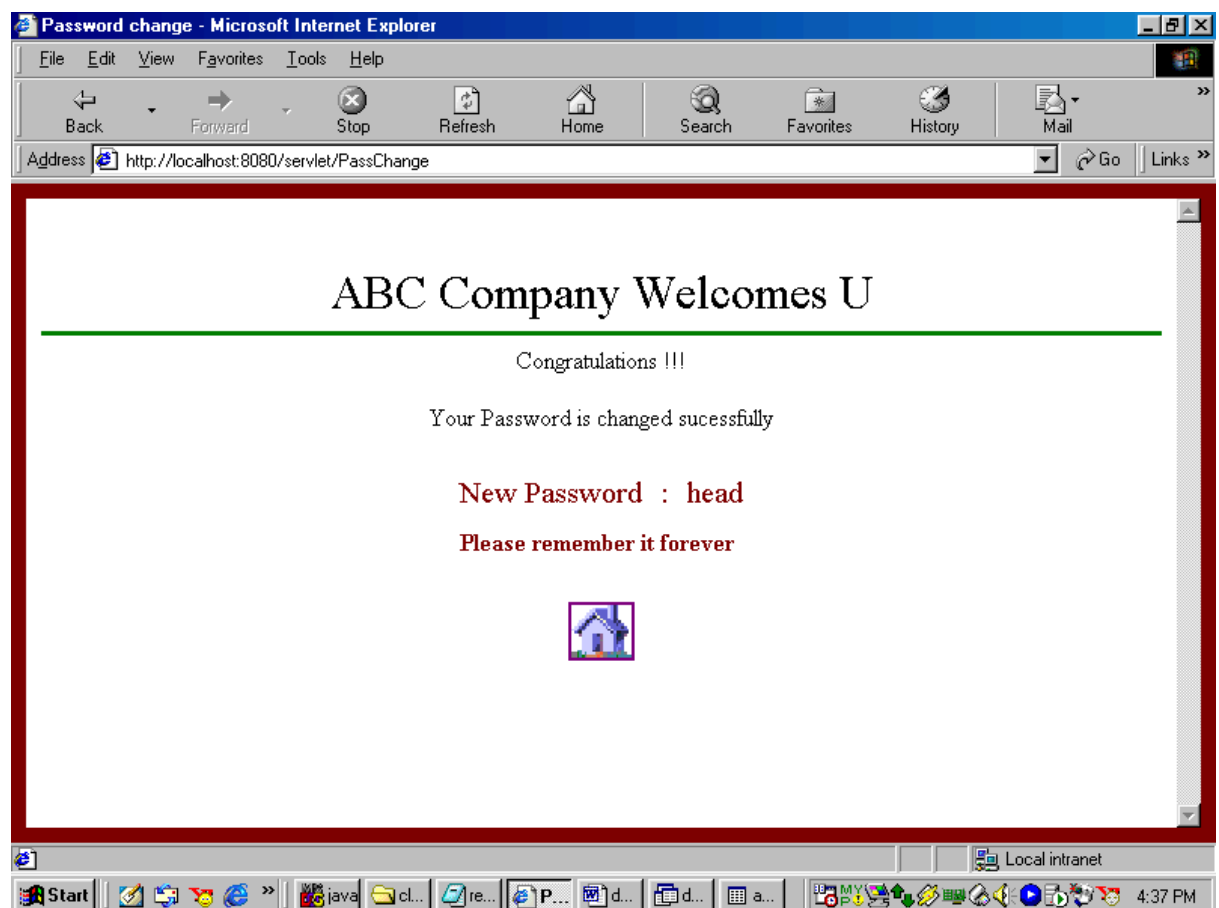
Old Password :

New Password :

Confirm New Password :

Done Local intranet

Start java cl... re... P... d... d... a... 4:36 PM



Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/servlet/ManagerFrame> Go Links

Report Page

Sales Person Report

Area Wise Report

Product Wise Report

[Back](#)

Sales Rep's Sales Details

Distributor Code : d2

From : 28-11-1981 [Choose Date](#)

To : 27-11-1999 [Choose Date](#)

Product Code : Auto Accessories

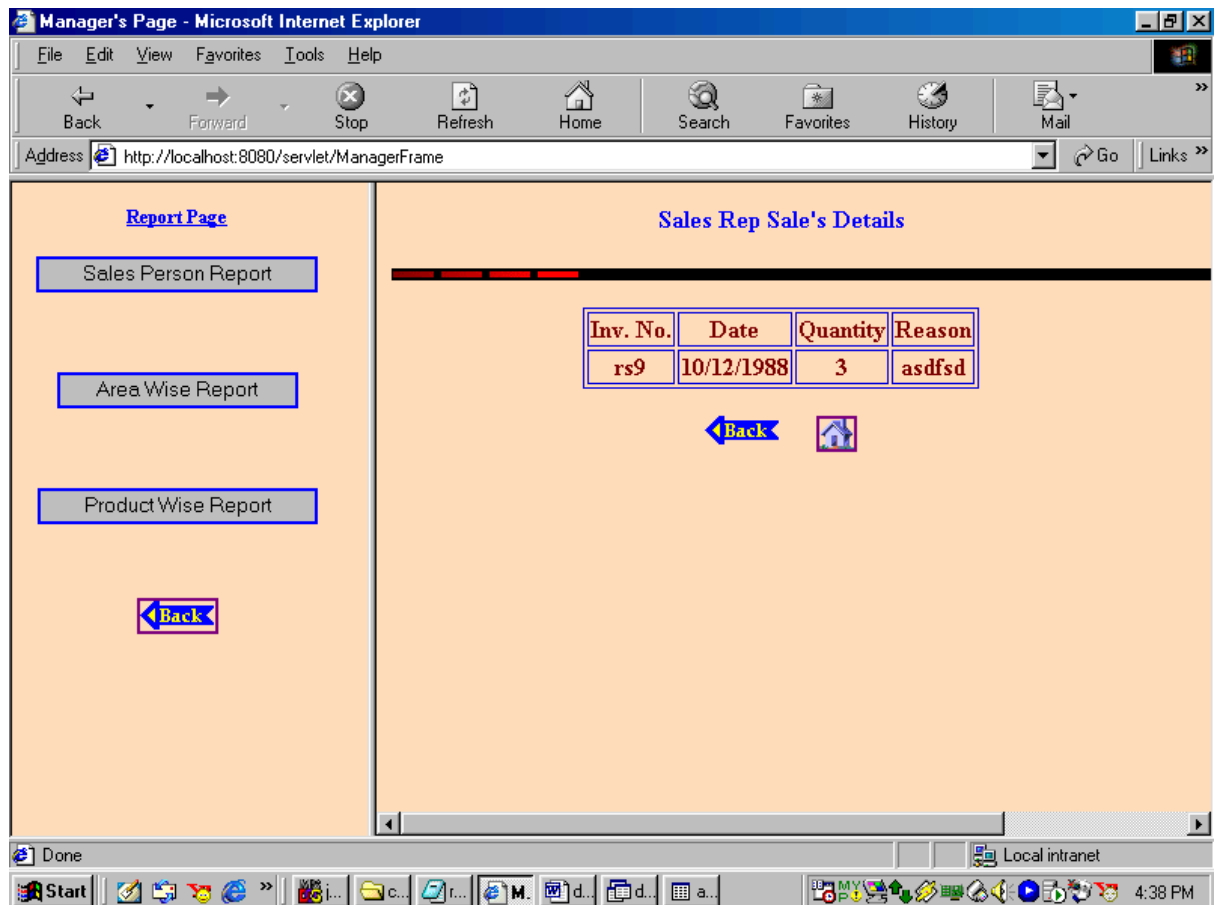
[Report](#)

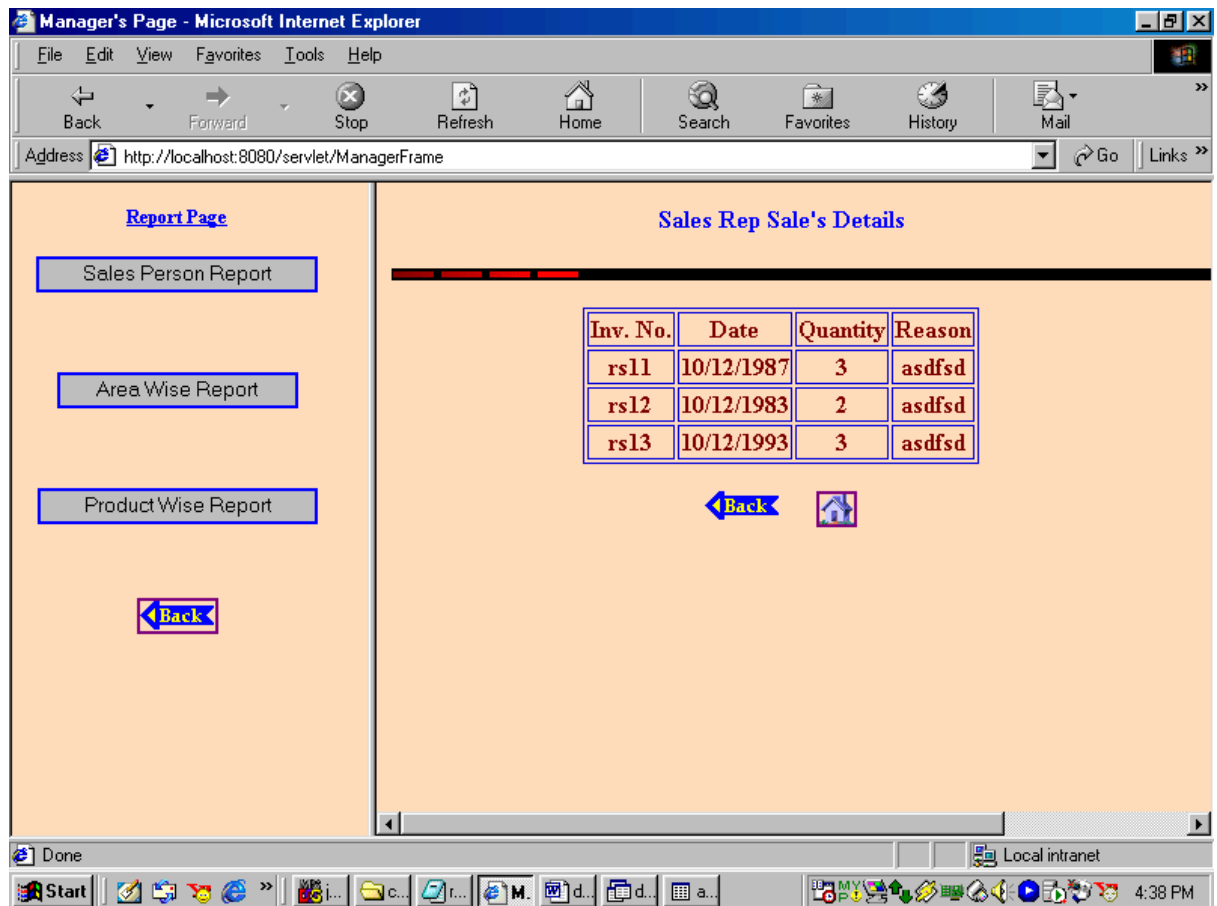
[Back](#) [Home](#)

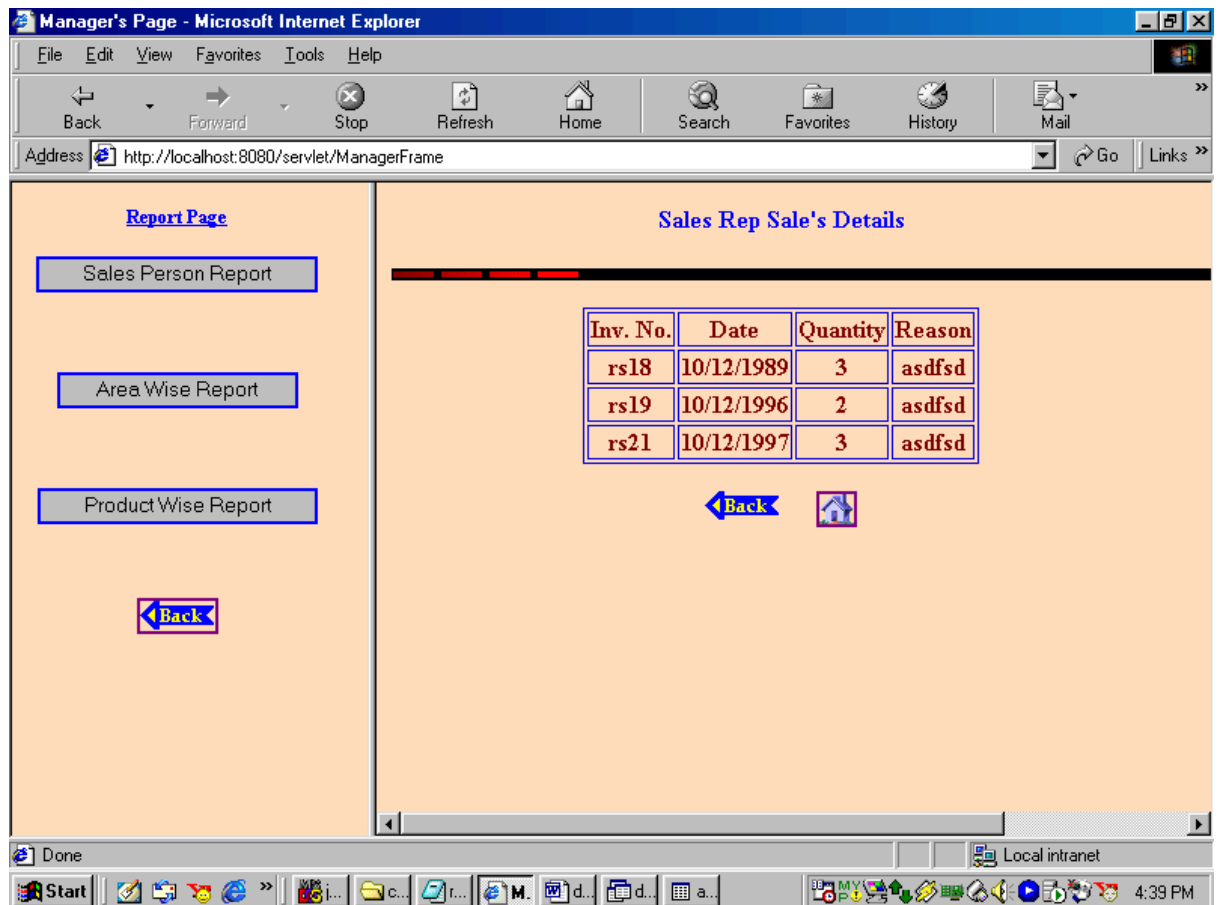
Local intranet

Start

4:38 PM







Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/servlet/ManagerFrame> Go Links

Report Page

Sales Person Report

Area Wise Report

Product Wise Report

[Back](#)

Sales Rep's Sales Details

From : 30-11-1980 [Choose Date](#)

To : 30-11-2007 [Choose Date](#)

Product Code : Auto Accessories

[Report](#)

[Back](#) [Home](#)

Local intranet

Start

WPS: i... C... r... M. d... d... a...

MY P... 4:41 PM

Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/servlet/ManagerFrame> Go Links

Report Page

Sales Person Report

Area Wise Report

Product Wise Report

[Back](#)

Sales Rep Sale's Details

Product Auto Accessories's Percentage Report

Sl.No	Rep Code	Area Code	Quantity	Percentage
1	rs1	d1	3	4.6153846%
2	rs2	d1	2	3.0769231%
3	rs3	d1	1	1.5384616%
4	rs4	d1	3	4.6153846%
5	rs5	d1	3	4.6153846%
6	rs6	d2	2	3.0769231%
7	rs7	d2	3	4.6153846%
8	rs8	d2	1	1.5384616%
9	rs9	d2	3	4.6153846%
10	rs10	d3	2	3.0769231%
11	rs11	d3	3	4.6153846%
12	rs12	d3	2	3.0769231%

Done Local intranet

Start [Taskbar Icons] 4:41 PM

Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/servlet/ManagerFrame> Go Links

Report Page

Sales Person Report

Area Wise Report

Product Wise Report

[Back](#)

Sales Rep's Sales Details

From : 23-11-2002 [Choose Date](#)

To : 30-11-2007 [Choose Date](#)

Product Code : Auto Accessories

[Report](#)

[Back](#) [Home](#)

Done Local intranet

Start [Taskbar Icons] 4:43 PM

Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address http://localhost:8080/servlet/ManagerFrame

[Report Page](#)

Distributor Report

Product Wise Report

[Back](#)

Distributor's Product Sale's Details

Distributor's Product Wise Report

Sl.No.	Inv. No.	Date	Quantity
1	s1	05/01/1997	1
2	s3	22/02/1984	1
3	s3	22/02/1988	1
4	s3	22/02/1995	1

[Back](#)

Done Local intranet

Start 4:44 PM

Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address http://localhost:8080/servlet/ManagerFrame

[Report Page](#)

Sales Person Report

Area Wise Report

Product Wise Report

[Back](#)

Sales Rep Sale's Details

Product Auto Accessories's Percentage Report

Sl.No	Rep Code	Area Code	Quantity	Percentage
1	rs7	d2	3	23.076923%
2	rs8	d2	1	7.6923075%
3	rs16	d4	1	7.6923075%
4	rs25	d7	3	23.076923%
5	rs26	d7	1	7.6923075%
6	rs27	d7	3	23.076923%
7	rs28	d7	1	7.6923075%

[Back](#)

Done Local intranet

Start 4:42 PM

Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/servlet/ManagerFrame> Go Links

Report Page

Distributor Report


Product Wise Report

[Back](#)


Distributor's Product Sale's Details

Distributor's Product Wise Report

SL.No.	Inv. No.	Date	Quantity
1	s1	05/01/1997	1
2	s3	22/02/1984	1
3	s3	22/02/1988	1
4	s3	22/02/1995	1

[Back](#) 

Done Local intranet

Start  4:44 PM

Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address http://localhost:8080/servlet/ManagerFrame

[Report Page](#)

Distributor Report

Product Wise Report

[Back](#)

Distributor's Sales Details

Distributor Code : d2

From : 10-11-1978 [Choose Date](#)

To : 25-11-1999 [Choose Date](#)

Product Code : Auto Accessories

[Report](#)

[Back](#) [Home](#)

Done Local intranet

Start 4:45 PM

Manager's Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address http://localhost:8080/servlet/ManagerFrame

[Report Page](#)

Distributor Report

Product Wise Report

[Back](#)

Distributor's Product Sale's Details

Distributor Sale's Report

Sl.No.	Inv. No.	Date	Quantity
1	s3	22/02/1984	1
2	s3	22/02/1988	1
3	s3	22/02/1995	1

[Back](#) [Home](#)

Done Local intranet

Start 4:44 PM

Registration Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/dms/dislogin.html> Go Links

New Distributor's Application Form

Login :

User ID : * Password : * Confirm Password : *

kannan ***** *****

Personal :

Name : * Address : *

vijaykannan raja bhathar vettri nagar

City : * State : * Pin Code : *

chennai tn 2342323

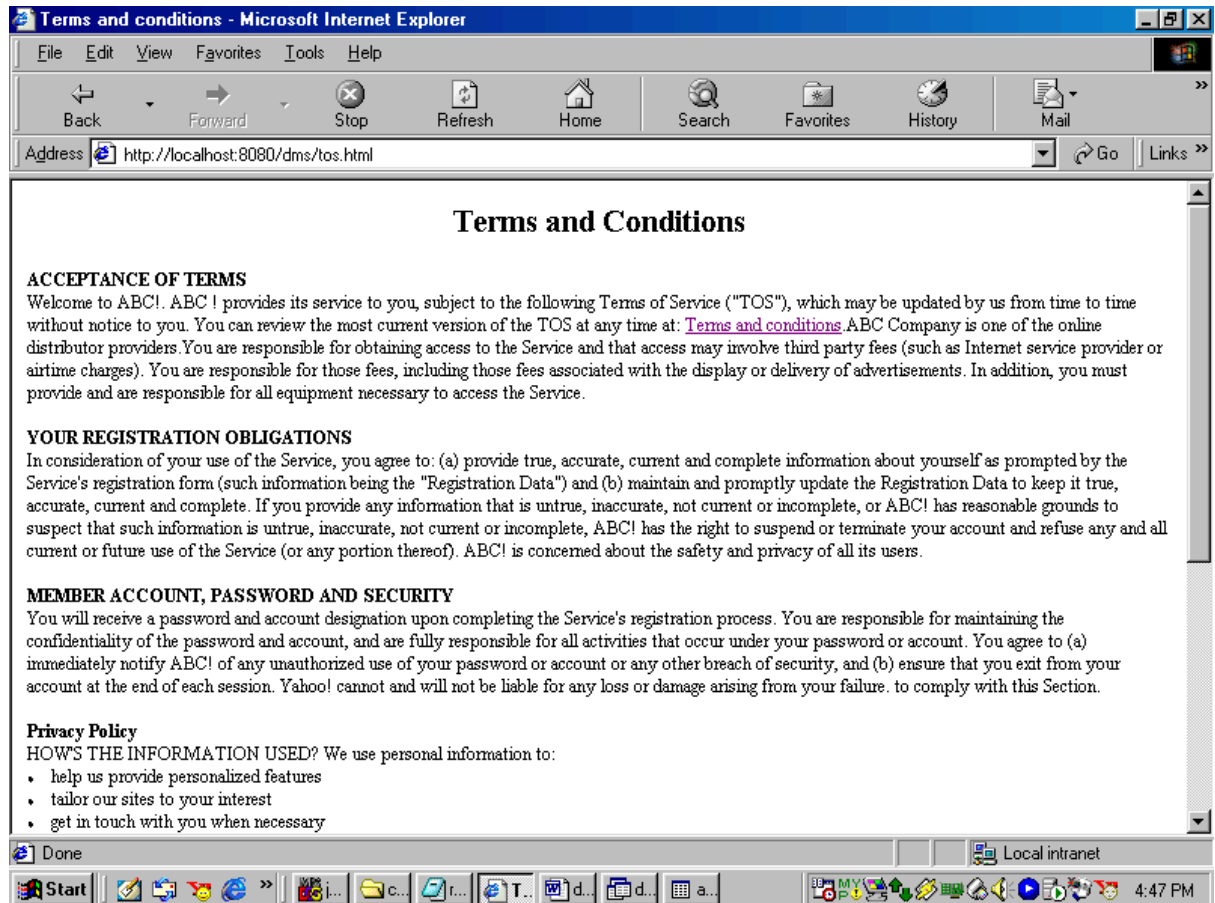
E-Mail ID : * Resi-Phone : * Work-Phone : *

vj@sify.com 3453454 345345

Continue

Done Local intranet

Start [Taskbar Icons] 4:47 PM



Registration Page - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Refresh Home Search Favorites History Mail

Address <http://localhost:8080/dms/disareacredit.html> Go Links

New Distributor's Application Form

Distributor Allotment Details:

State :

District :

City :

Area :

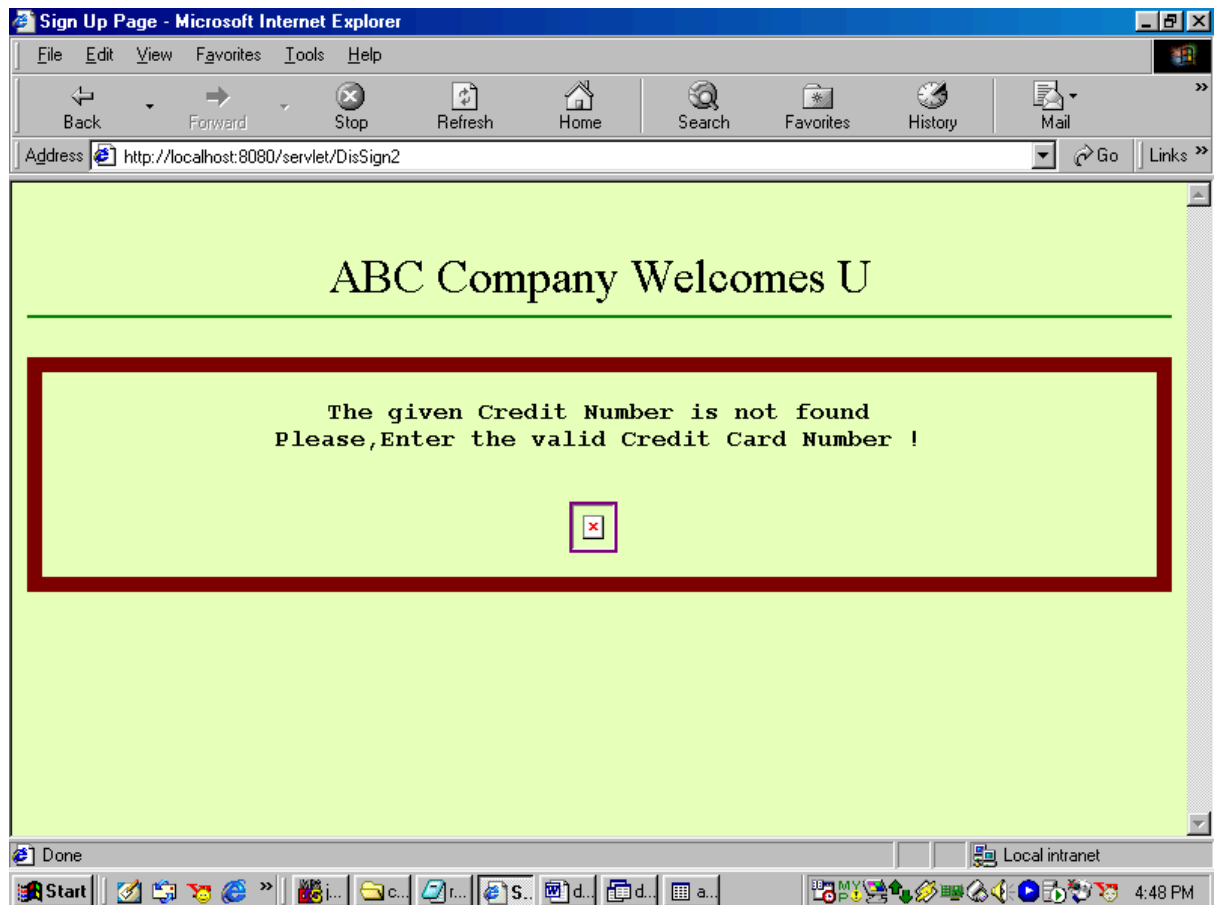
Credit Card Details

Credit Card No. :

Bank Name :

Done Local intranet

Start [Taskbar Icons] 4:48 PM



13. CONCLUSION

The system is developed for automating the activities involved in the sales done by the company. The data maintenance and reports generated are compatible with the daily activities of the department.

The main advantage of the system over the existing system is the increase in the speed of information retrieval since that data is maintained

systematically and any type of information required for the end user of the system can be generated easily.

This system is of vital use to the sales incharge, to the distributor and to the representative also. They can very well know about the sales done by them on a particular day or a month, between dates, in an area,...

Though "**Distributor Information System**" is so designed that not much amendment is required. This system has provisions for adapting to future enhancements.

14. BIBLIOGRAPHY

- 1. Roger S.Pressman ,“Software Engineering : A Practitioners Approach “ ,
Fourth Edition, McGraw-Hill companies inc., New York 1997.**

2. **RICHARD FAIRLEY**, “Software Engineering Concepts”, **Fourth Edition**, **McGraw-Hill Companies inc.**, **New York, 1997.**
3. **Patrick Naughton and Herbert Schildt** , “The Complete Reference Java 2”, **Third edition**, **The Tata McGraw-Hill Edition 1999.**
4. **Java™ 2 SDK**, Standard Edition Documentation Version 1.4.0
5. **Dustin R. Callaway**, “Inside Servlets : Server-Side Programming for the Java™ Platform”, **Addison - Wesley**
6. **Karl Moss**, “Java Servlets”, **Second Edition**, **The McGraw-Hill companies inc.**, **Newyork 1999.**
7. **Eric Donald and O'Donnell**, ”Using Html 4, Xml and Java1.2”, **Printice-Hall of India, 1999.**