

P-1911

# Generic ERP For Material Management

By

**Santhosh.S**

**Reg. No. 71204621035**

Of

**KUMARAGURU COLLEGE OF TECHNOLOGY  
COIMBATORE**

**A PROJECT REPORT**

Submitted to the

**FACULTY OF INFORMATION AND COMMUNICATION ENGINEERING**

In partial fulfillment of the requirements

for the award of the degree

of

**MASTER OF COMPUTER APPLICATIONS**

**July, 2007**



P-1911

### Bonafide Certificate

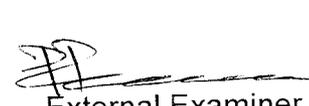
Certified that this project report titled **Generic ERP For Material Management** is the bonafide work of **Mr. Santhosh.S** who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other project report or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

  
Project Guide

  
Head of the Department

Submitted for the University Examination held on 31/7/07

  
Internal Examiner

  
External Examiner 3/7/07

MNCL/ADMIN/650/2007

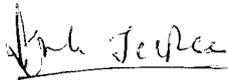
13/05/2007

**TO WHOMSOEVER IT MAY CONCERN**

This is to inform that Mr. Santhosh.S (71204621035) final year MCA student of Kumaraguru College of Technology , Coimbatore has successfully completed his Project work titled “**Generic ERP for Material Management**” during the period of December 2006 to May 2007 .

We wish him all success for his future endeavors.

For Metronet Communication,



Krupa Japee  
Administrator.

## Acknowledgement

I express my grateful thanks to our beloved principal, **Dr. Joseph V Thanikal** Kumaraguru College of Technology, Coimbatore, for giving me an opportunity to take up this project.

I express my deep sense of gratitude to **Dr.S.Thangasamy**, Professor and Dean of Computer Science and Engineering department and **Dr.M.Gururajan**, Professor and Head of Department of Computer Applications for extending their help in providing all the facilities at college for the successful completion of the project.

I also thank **Mr.A.Muthukumar**, Assistant Professor and course co-ordinator, Computer applications, for providing all the desired documents and details regarding the various aspects necessary to the proper functioning of the system.

I would like to express my deepest and sincere gratitude to **Mr M.Manikandan**, Senior Lecturer, Department of Computer Applications, for his guidance, support, cooperation and valuable suggestions during the course of this project.

## SYNOPSIS

The purpose of the project is to provide a Enterprise resource planning package for a industry that has various operations such as marketing, sales, warehouse, purchase, quality management, research and development, sales department. Whenever a product is entered the product is saved in the database and is sent into warehouse department, the warehouse department then checks for the quality management and the product is then entered into the manufacturing department and then the product is deliverable to the customer. The project gives a user interface between the user and the system by agent controls, so that it is easy for the system providing flexibility.

## TABLE OF CONTENTS

Topic	Page No.
Abstract	iii
List of Tables	vi
List of Figures	vii
1. Introduction	1
1.1 Organization Profile	1
1.2 Project Profile	2
2. System Study and Analysis	3
2.1 Literature Review	3
2.1.1 Existing System	3
2.1.2 Drawbacks of the Existing System	3
2.1.3 Proposed System	4
2.1.4 Need For Proposed System	4
2.2 Problem Specification	4
2.3 System Specification	7
2.3.1 Hardware Configuration	7
2.3.2 Software Configuration	7
2.4 Software Specification	8
3. System Design	14
3.1 Logical Design	15
3.1.1 Data Flow Diagram	15
3.2 Physical Design	19
3.2.1 Input Design	19
3.2.2 Output Design	19
3.2.3 Database Design	20

4. System Testing and Implementation	21
4.1 System Testing	22
4.2 System Implementation	24
4.2.1 Implementation Procedure	24
4.2.2 Operational Documentation	26
4.3 Quality Assurance	27
4.3.1 Generic Risks	27
4.3.2 Intolerable	27
4.3.3 ALARP	28
4.3.4 Acceptable	28
4.3.5 Hazard Probability Reduction	28
4.3.6 Accident Prevention	28
4.4 System Security	29
5. System Maintenance	30
6. Conclusion and Future Enhancement	32
6.1 Conclusion	32
6.2 Future Enhancement	32
Appendices	51
Reference	55

## List of Tables

<b>Table Description</b>	<b>Page No</b>
Table A 1.1 Admin Table	33
Table A1.2: Marketing Table	33
Table A1.3: Research and Develpoment Table	34
Table A1.4: Warehouse Table	34
Table A1.5: Purchase Order Table	35
Table A1.6: Quality Table	35
Table A1.7: Manufacturing Table	35
Table A1.8: Sales table	36

## List of Figures

	<b>Figure Description</b>	<b>Page No</b>
Figure 4.3.3	Login Screen	37
Figure 4.3.4	Marketing	38
Figure 4.3.5	Research and Development	39
Figure 4.3.6	Warehousing	40
Figure 4.3.7	Purchase	41
Figure 4.3.8	Quality	42
Figure 4.3.10	Manufacture	43
Figure 4.3.11	Sales	44

## CHAPTER 1

### INTRODUCTION

#### 1.1 Organization Profile

Metronet Microsystems was formed in 1999. In response to the following market conditions: MetroNet Microsystems is a Product development company. MetroNet Microsystems have entered into nanotechnology development products. MetroNet Microsystems products visions are towards Wireless Networking, Embedded systems, DSP, Networking WCDM, 3G, CDMA areas. Keeping the long term growth objectives in mind, MetroNet Microsystems evolved business relationships with the leading companies in providing software solutions. It plans to address both overseas and the Indian Market.

MetroNet Microsystems have established relationship with clients, both in India and abroad who often contact us for developing projects. MetroNet Microsystems keeps in touch with emerging and growing companies. MetroNet Microsystems presents their abilities and achievements to someone in a position not only to fully appreciate them. MetroNet Microsystems are not part of a network which allows others access to their identity / information, their identity will never be shared with any one except a potential employer. The services offered are: Software Development – Off Shore / On-Site on Java, .Net, Telecom, DSP and Embedded Systems.

- Product on DSP, Micro controller and Embedded Systems
- IT Enable Services
- Product Related Solutions

## 1.2 PROJECT PROFILE

This project ENTERPRISE RESOURCE PLANNING is purely based on the Management of an effective Organization. It deals with passing information between the departments for interaction between departments. The Departments in an Organization are,

- Marketing Department
- Research And Development Department
- Warehouse Department
- Purchase Department
- Quality Department
- Manufacture Department
- Sales Department

These are the various departments the main objective of this project is to computerize the activities of these departments and to provide effective communication between departments.

## CHAPTER – 2

### SYSTEM ANALYSIS

#### 2.1 LITERATURE REVIEW

##### 2.1.1 Existing System

The existing system maintains the records manually. There will be no proper integration between the Departments because the employees will not interact with each other. There will be no proper transfer of records between the Departments. Some records may be lost due to improper maintenance.

##### 2.1.2 Drawbacks of Existing System

The Existing System has the following drawbacks,

- We cannot give the guarantee because data in papers can be erased easily at any time.
- It takes long time to find any employee details.
- There may be a chance of error occurrence at the time of entering the data.
- As it is done manually, it becomes a time consuming process.
- It is difficult to prepare reports.
- It is difficult to find the completed reports.
- And also, Maintenance of reports is tough task because there may be chance for missing data.

### **2.1.3 Proposed System**

The proposed system is based on computerizing the existing manual system. In this system the department activities are computerized and the information are passed between the departments for effective communication.

### **2.1.4 Need for Proposed System**

The proposed system has the following benefits,

1. This system does not want a huge number of notebooks and stationeries.
2. Searching process is easier.
3. Only the Administrator can do the modifications.
4. Reports can be easily generated.
5. There is no chance of error occurrences during the time of entering the data.
6. It will reduce the Time.

## **2.2 PROBLEM SPECIFICATION**

The Enterprise Resource Planning is important for all large organizations. It is used for finding the profit or loss in an organization with the proper maintenance of records and proper interaction with in an organization. There are six main modules, which are developed to accomplish the organizations objectives. The modules are,

- Marketing Management Module
- Research & Development Module
- Quality Management Module
- Warehouse Department Module
- Purchase Order Management Module
- Manufacturing Department Module
- Sales Department Module

### **Marketing Management Module**

Marketing Management Module is the primary module that interacts with the client's request as the marketing person gets the new order for the product through phone calls, e-mail and executives. The client's request are stored in database. The Research & Development department will be informed about the client's request.

### **Research & Development Module**

The R&D Department Module will work on the project and find out the requirements of the product. The requirements will be sent to Warehouse to verify the availability of products. The rejected product from the Quality department will also be sent back to the R&D department.

### **Warehouse Department Module**

The Warehouse Department Module will maintain the stock of parts. It will send the request to the Purchase Department if the parts are not available.

### **Purchase Order Management Module**

Purchase Order Management Module department is responsible for the purchase of the parts/products required by client as per the information provided by the Warehouse Department. The purchase products are send back to the Warehouse Department.

### **Quality Management Module**

The Quality Management Module is the most important of all departments which satisfies the client's need. This department will check the quality of the product. If the quality is approved then the company will fix rate for the product and send to Manufacturing Department to develop the remaining products. Else it will send the product to R&D Department to inform that the quality does not satisfies the client's request.

### **Manufacturing Department Module**

The Manufacturing Department Module will work on the client's request to develop the product needed. It will develop the remaining products once when the product is approved by the Quality Department.

### **Sales Department Module**

The Sales Department Module is the final module which interact with client's to provide them the final product as a fulfillment of their requirements with in the stipulated period.

## 2.3 SYSTEM SPECIFICATION

### 2.3.1 Hardware Configuration

Processor	:	Intel Pentium IV 1.7GHz
RAM	:	256 MB SDRAM
Hard Disk	:	40 GB
Floppy Disk	:	1.44 MB
Monitor	:	VGA Color
Modem	:	56 Kbps

### 2.3.2 Software Configuration

Operating System	:	Windows XP, Windows 2000
Application Software	:	Internet Information Server
Front End	:	ASP.NET
Back End	:	SQL-Server

## **2.4 SOFTWARE SPECIFICATION**

### **ASP.NET**

The .Net Framework is the heart of the Microsoft .NET Platform. And the foundation of .NET is the Common Language Runtime (CLR). It is the CLR that performs the tasks of compiling the source code, managing basic security, garbage collection, general memory management and other base-level services.

Along with the CLR, the Framework Class Library (FCL) provides the function and services needed to build a fully functional application. The FCL contains hundreds of classes that can be collected together in related groups.

Major Components of the .NET Framework is the .NET Framework is the .NET Framework class library, a set of object-oriented, reusable classes that can be used to develop complex distributed, dynamic applications with ASP.NET.

Asp.NET offers a new way to build web applications with “smart” web forms and powerful web services.

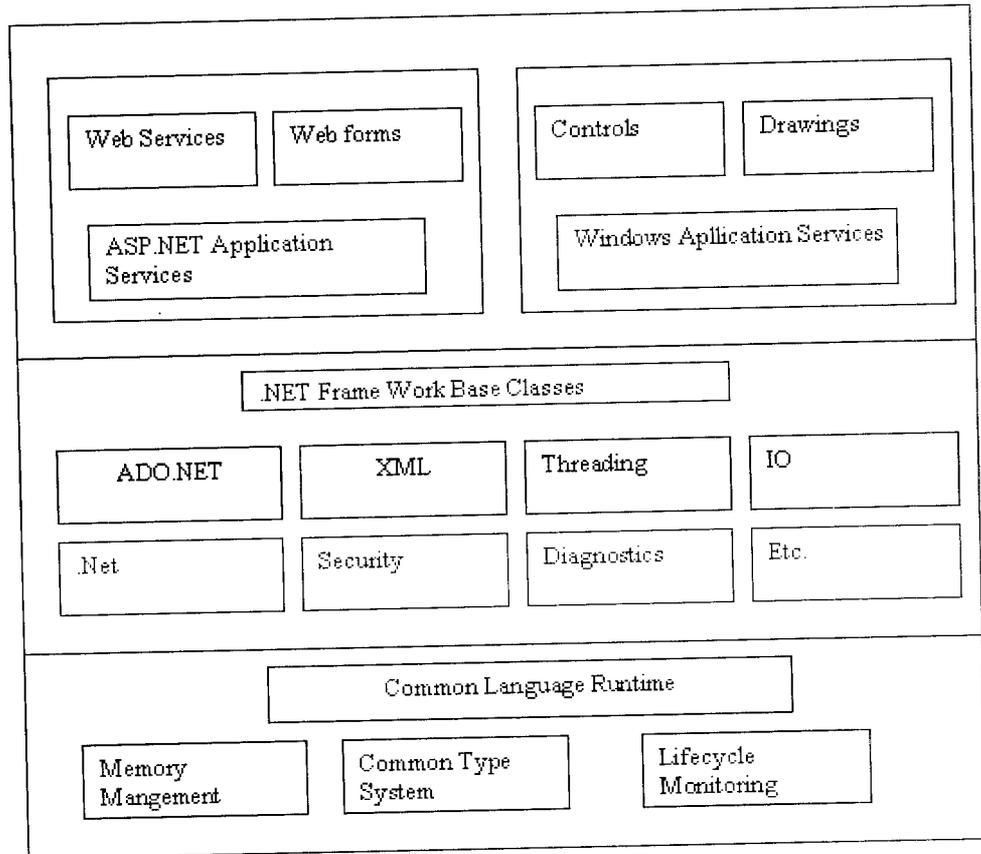


Figure 2.1 .Net Framework

### ASP.NET Web Forms

ASP.NET offers a new, evolutionary approach to build web applications on the Microsoft platform. This includes an improved, object-oriented programming model for creating web pages, built in support for multiple model for creating web pages and clear separation between markup and execution code.

### Improved Web Page Authoring

A Microsoft Web author interpreted both client and server side scripting on the same page. Also, programmers had to constantly switch between server-side and client-side coding in order to create fully functional solutions.

## Simplified Programming Model

ASP.NET supports a complete language mode including Visual Basic, C# and many other third-party languages. In addition, ASP.NET solutions are fully compiled on the server. The results are fully compiled on the server. The results are increased performance and improved reliability.

## Multiple Client Support

The ASP.NET Web Forms model also provides automatic support for multiple client browsers and devices.

## ASP.NET Web Services

The ability to build SOAP (Simple Object Access Protocol) based web remote procedure invocations called Web Services. The Web Service feature of ASP.NET lets programmers publish components via the Internet.

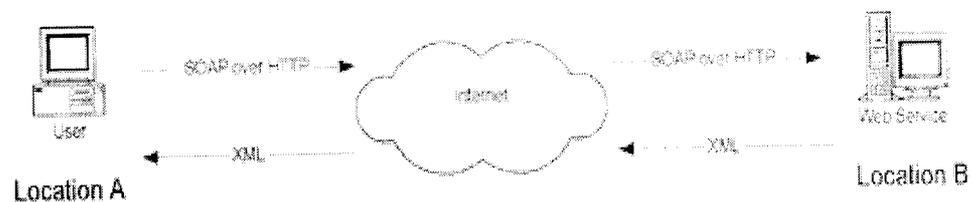


Figure 2.2 .Net Web Services

## Easy Programming Model

While the SOAP interface is powerful, it also involves a complex of XML, protocol stacks and message handling at the sockets level.

## ASP.NET Infrastructure

ASP.NET also offers a number of infrastructure innovations that make ASP.NET more secure, easier to configure and deploy and more scalable and reliable.

A few of the more important improvements in the ASP.NET are,

- Powerful Security Model
- Improved Deployment and update.
- Easier Configuration
- Increased Scalability and Availability
- Automatic Restarts

## ASP.NET Features

- Simplified Programming Model
- Simplified Deployment
- Better Performance
- Web Services
- Security
- Better Session Management
- Code Behind Pages
- Simplified Form Validation
- Powerful Controls

## Features of C#

C# is a modern programming language. The programming language `c#` is derived from C and C ++. There are features they are,



P-1911

## **Simple**

The foremost goal of C# is simplicity. Pointers are a prominent feature that is missing in C#. C# provides with a unified type system. This type system enables us to view every type as an object.

## **Modern**

Decimal data type is provided for memory calculations. Data types can be specifically created for any type of application.

## **Object-Oriented**

C# supports all the key object-oriented concepts such as encapsulation, inheritance and polymorphism. There are no global function, variables and constants. C# supports the private, protected, public modifiers. It allows one base class and the multiple inheritance is implemented using inheritance.

## **Type-Safe**

C# implements strict type safety by following some rules,

- Uninitialized variables cannot be used.
- An integer cannot be cast to a reference type.
- It performs bounds checking on array elements.
- Overflow checking is enabled.

## **Compatible**

C# allows access to different APIs, with Common Language Specification (CLS). C# supports OLE Automation.

## **Flexible**

C# is a most flexible language. C# code runs in safe mode.

## **Feature of SQL SERVER 7.0**

SQL Server is a SQL based client server database. SQL based is that it is used in ANSI (American National Standard Institute) version of structured query language. SQL Server is a collection of many different components. This includes SQL Server agent, SQL Mail and distributed transaction co-ordinates.

There are some features they are,

- Dynamic self-management like memory and locking.
- Very large database support.
- More flexible replication.
- OLE DB support.
- Parallel query support.

## CHAPTER 3

### SYSTEM DESIGN

The most creative and challenging phase of the system development process is design phase, it is a solution, a “how to” approach to the creation of the proposed system. Design, the first step in the development of an engineered product is initiated only after a clear exposition of expected product functions becomes available. The design of a software system and its component parts should follow an ordinary sequence of steps. The design process continues by means of stepwise refinement through a series of abstraction levels until a physical realization replaces the abstraction. A design can also be regarded as a detailed plan for a solution to a problem certainly it has been demonstrated over and over that simplicity is at the heart of design elegance. Simple designs are easily understood, easily built, and easily tested. Simplicity is the most important criteria of a design. Other design criteria include the following.

**Documentation:** A good design always comes with a set of well-written documents.

**Testability:** In a good design every requirement is testable. A design that cannot be easily tested against its requirements is not acceptable design.

**Structure:** A good design presents hierarchical structure that makes logical use of control policies among components.

**Modularity:** a good design is modular and exhibits the properties of high cohesiveness and low coupling.

**Discreteness:** A good design separates data procedures and timing consideration to the extent possible.

**Representation:** A good design should be easily communicated to all interested parties through appropriate abstraction and representation.

**Reusability:** A good design should be repeatable and reusable.

### 3.1 LOGICAL DESIGN

Logical design identified the records and relationships to be handled by the system. It focuses on the logic, or the reasoning behind the system or breaking down the system into subsystem. The logical design establishes the relationships among the various subsystems, the records and variables into the subsystems.

#### 3.1.1 Data Flow Diagram

The data flow diagram (DFD) is one of the most important tools used by system analysts. Data flow diagrams are made up of a number symbols, which represent system components. Most data flow modeling methods use four kinds of symbols. These symbols are used to represent four kinds of system components. Processes, data stores, data flows and external entities. Circles in dfd represent processes. Data Flow represented by a thin line in the DFD and each data store has a unique name and square or rectangle represents external entities.

To construct a Data Flow Diagram, we use,

- Arrow
- Circles
- Open End Box
- Squares

An arrow identifies the data flow in motion. It is a pipeline through which information is flown like the rectangle in the flowchart. A circle stands for process that converts data into information. An open-ended box represents a data store, data at rest or a temporary repository of data. A square defines a source or destination of system data.

#### Six rules for constructing a Data Flow Diagram

- Arrows should not cross each other.
- Squares, circles and files must bear names.
- Decomposed data flow squares and circles can have same names.
- Choose meaningful names for data flow
- Draw all data flows around the outside of the diagram.

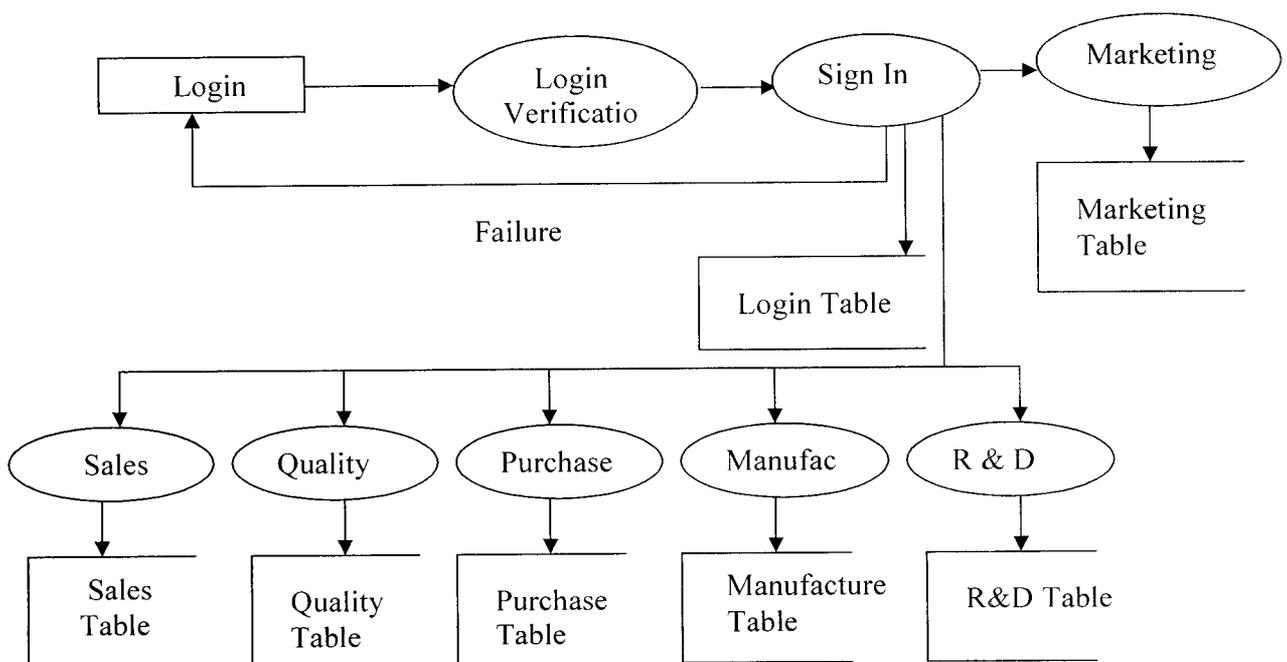


Figure 3.1 DFD for Enterprise Resource Planning

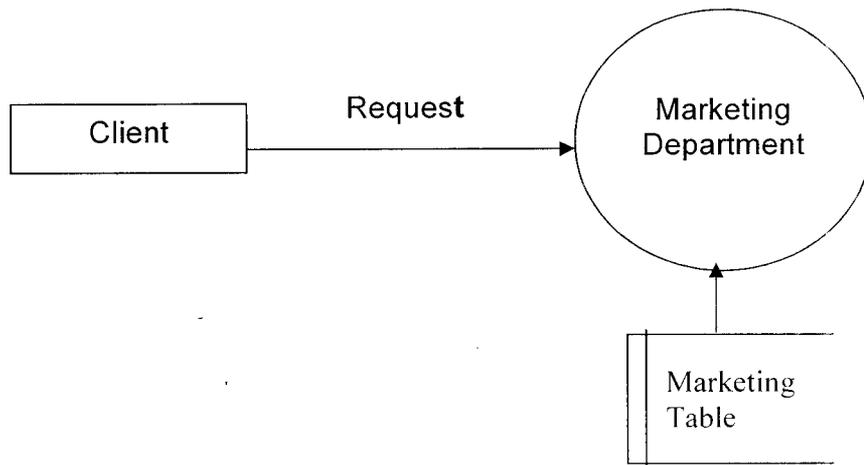


Figure 3.2 DFD for Marketing

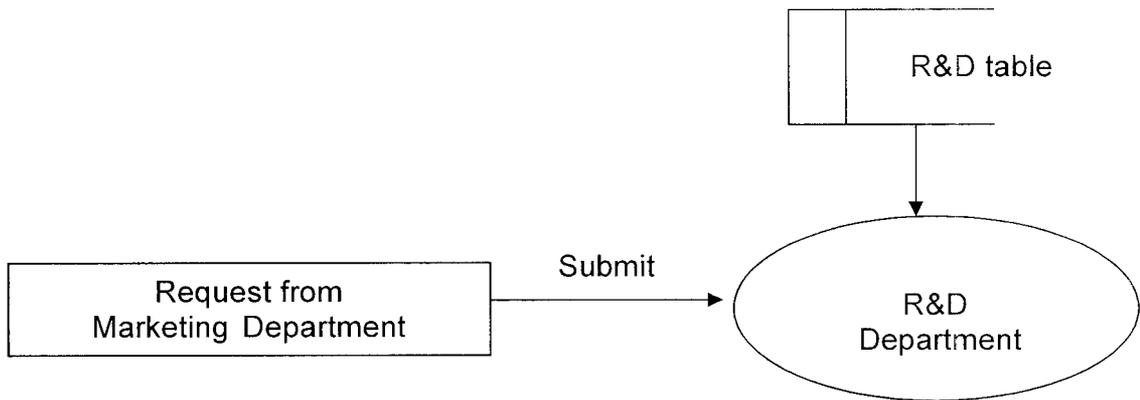


Figure 3.3 DFD for R & D

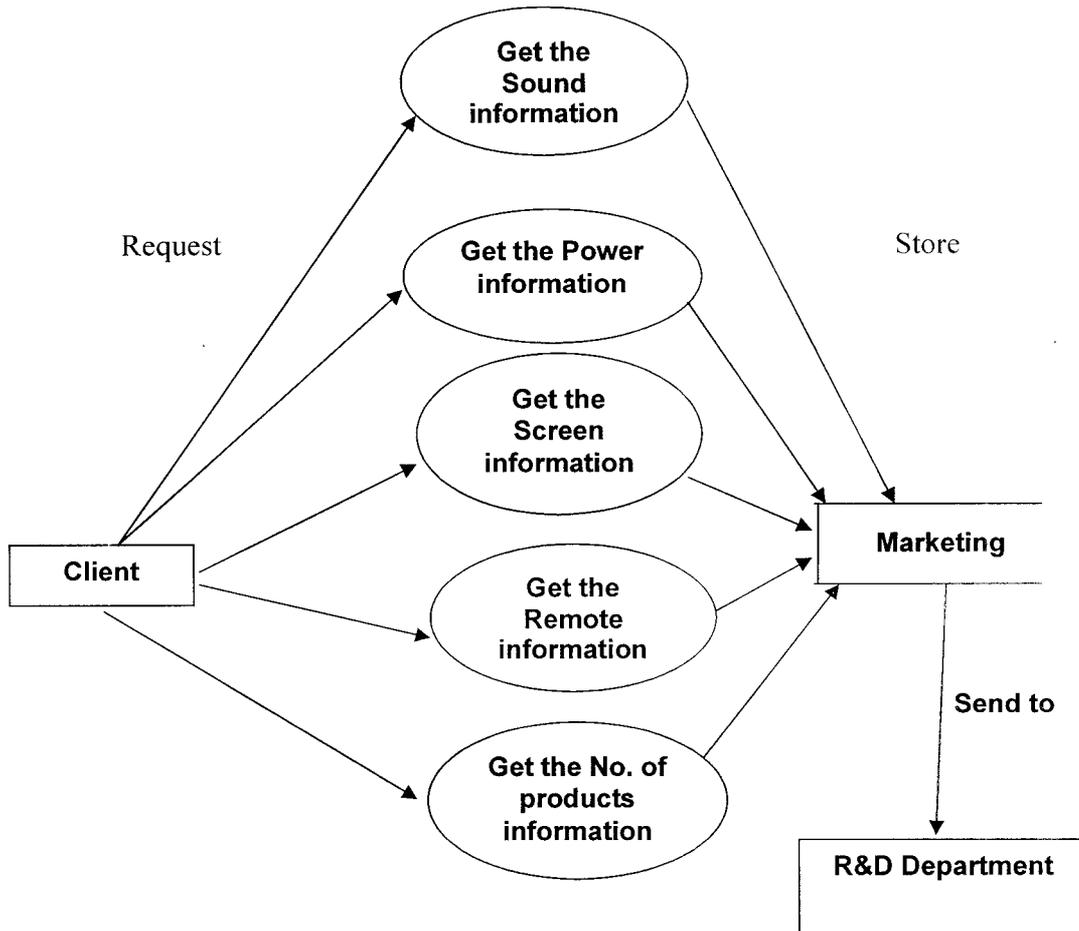


Figure 3.4 DFD for Marketing Process

## 3.2 PHYSICAL DESIGN

### 3.2.1 Input Design

Input design is the process of converting user originated inputs to a computer based format. Input design is the part of the System design if it is incorrect then the processing and output will magnify the errors. Inaccurate input data is the

most common cause of errors in data processing. The main objective of designing input focus on,

- Controlling the amount of input required.
- Avoiding delayed response.
- Controlling errors.
- Keeping process simple.
- Avoiding errors.

The required inputs are stored in the form of tables. They may be numeric and alphanumeric input screen should be user friendly, so that every one can access the options without having the complete system knowledge.

### **3.2.2 Output Design**

The output must be provided in such a format that the people can understand. After analyzing the operations of the systems, output information required for each jobs are determined. In addition to this, these outputs may be in format suitable to input for subsequent processing.

A major form of output is a hard copy from the printer. Printout should be designed around the output requirements of the user. Efficient intelligible output design should improve the system relationship with user. Output design refers to the results generated by the system. The output of a system can take many forms. The most common forms are reports, screen displays, printed form and graphical drawing forms.

The normal procedure in developing a system is to design the output in detail first and then move back to the input. The output will be in the forms of views and reports.

### 3.2.3 Database Design

A database is a collection of inter-related data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quickly, inexpensive and flexible for the user. While designing the database, redundancy should be avoided to server many users who use the database effectively and also to increase the system performance.

It minimizes the artificiality embedded in using separate files. The primary objectives are fast response time to enquiries, more information at low cost, redundancy control, clarity and ease of use, accuracy and fast recovery. Database design is used to define and then specify the structure of business used in the client/server system. A business object is nothing but information that is visible to the users of the system. The database must be normalized one.

#### Data Normalization

The entities along with their attributes can be stored in many different ways into a set of tables. The methods of arranging these attributes are called normal forms. The theory behind the arrangement of attributes into table is known as normalization theory.

It helps in

- Minimization of duplication of data.
- Providing flexibility to support different functional requirements.
- Enabling the model to be translated to database design.

All relations in a relational database are required to satisfy the following condition:

Every value in a relation - each attribute value is atomic - non decomposable so far as system is concerned.

Advantages of normalization are:

- Helps in reduction in the complexity of maintaining data integrity by removing the redundant data.
- It reduces inconsistency of data

### **First normal form**

- 1 Eliminate the repeating fields.
- 2 Creates a row for each occurrence of a repeated field
- 3 Allows exploitation of column functions

For example consider the demand table. Here all the fields that are found to be repeating are given a row. The repeating fields are thus eliminated. The first normal form is carried out while designing the tables itself.

### **Second Normal Form**

The second normal form has the characteristics of the first normal form and all the attributes must fully be dependent on the primary key.

## CHAPTER 4

### SYSTEM TESTING AND IMPLEMENTATION

#### 4.1 SYSTEM TESTING

System testing is executing a program to check logic changes made in it and with the intension of finding errors making the program fail. Effective testing does not guarantee reliability. Reliability is a design consideration.

The objective of testing is a process of executing a program with the intent of finding the errors. Testing provides the last option from which quality can be assessed and more programmatically, errors can be uncovered. Testing is an individualistic process and the number of different development approaches. System testing is actually a series of different development approaches. System testing is actually a series of different tests whose primary purpose is to fully exercise the computer-based system. Although each test has a different purpose, all work to verify that all system elements have been properly integrated and perform allocated functions.

Also the system testing is a process of checking whether the developed system is working according to the original objectives and requirements. System should be tested experimentally with the test data so as to ensure that the system works according to the required specification when the system is found working, test it with the actual data and check the performance.

System testing begins by testing program modules separately followed by testing bundled modules as a unit. The total system as a single unit is tested for recovery and feedback after various failures to ensure that no data are lost during emergency.

There are three major types of system testing they are,

- Unit Testing
- Validation Testing
- Integration Testing

### **Unit Testing**

Unit testing focuses verification effort on the smallest unit of software design that is the module. The unit testing is always whitebox oriented and the step can be conducted in parallel for modules. The module 'interface' is tested to ensure but the information properly flows into and out of the program unit test. The 'local data structures' are examined to ensure that data stored temporarily maintains its integrity during all steps in an execution.

Boundary Conditions' are tested to ensure that module operates properly at boundaries established to limit or restrict processing. All 'independent paths' though the control structure are exercised to ensure that all statements in a module have been executed at least once. Finally, all 'error handling paths' are tested.

### **Integration Testing**

Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high-order tests are conducted.

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing.

## **Validation Testing**

At the end of the integration testing, software is completely assembled as a package, interfacing errors have been uncovered and correction testing begins.

Software testing and validation is achieved through a series of black box tests that demonstrate conformity with the requirements. A test plan outlines the classes of tests to be conducted and a test procedure defines specific test cases that will be used to demonstrate conformity with the requirements. Both the plan and the procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are met.

After each validation test case has been conducted, one of the two possible conditions exists, they are

1. The function performance characteristics confirm to specification and are accepted.
2. A deviation from the specification uncovered and a deficiency list is created.

The deviation or error discovered at this stage in a project can rarely be corrected prior scheduled completion. It is necessary to negotiate with the customer to establish a method.

## **4.2 SYSTEM IMPLEMENTATION**

### **4.2.1 Implementation Procedure**

Implementation is the process of converting a new system into an operational one. The designed system is converted to an operational one using a suitable programming language.

Implementation includes all those activities that take place to convert an old system into new. Proper implementation is essential to provide a reliable system to meet the organizational requirement. The most commonly used implementation methods are pilot running and parallel running.

Processing the current data by a single user at a time is called the pilot running process. When one user is accessing the data at one system the system is said to be engaged and cannot be used by the user at another machine connected in network. This process is used in systems where more than one user is restricted. Processing the current data by more than a single user at a time is called the parallel running process. The same system can be viewed and accessed at the same time on different machines completed in network. This process is useful in systems where multiple users are entertained.

The implementation phase involves careful planning, investigation of the systems and constraints, design of methods to achieve the change over, the training of staffs in the change over phase and finally the evaluation of the change over. The online examination supports running procedures. It has been designed and developed for use by many users at a time. The examination is attended by many candidates at the same date and time. So more care is to be taken during the implementation of software.

There are three types of implementation:

- a. Implementation of computer system to replace a manual system. The problems encountered are converting files, training users, creating accurate files and verifying printouts for integrity.
- b. Implementation of a new computer system to replace an existing one. This is usefully a difficult conversion. If not

properly planned, there can be many problems. Some large computer systems have taken as long as year to convert.

- c. Implementation of a modified application to replace an existing one, using the same computer. This type of conversion is relatively easy to handle, provided there are no major changes in the files.

#### 4.2.2 Operational Documentation

Help and message systems are usually geared towards the end user of the system. The paper documentation that accompanies the system must also provide information for the technical staffs and engineers responsible for administering the system. So that it may help them in diagnosing and repairing system faults.

A software product is composed of code and documentation. Documentation consists of all the information about the software except the code itself. In size the code is by far the smaller part of the product. The production of effective documentation is sometimes overlooked, but it is vital to the success of software engineering. Documentation is aimed at 3 different audiences

- The software engineer: who will depend on documents from previous life cycle stages to guide continued development and maintenance.
- The manager: who will use documents from project past projects to plan and understand current projects.
- The user: Who learn how to use the software from documentation?
- It often seems unfair to the software engineer that a user judges the product on the basis of its documentation rather than the performance of the code. From the engineers limited viewpoint he fails to see that the best program in the world is useless if one does not know how to operate it.
- In preparing documentation, careful consideration has to be given to a number of factors.

- First the documentation should be complete.
- Second the documentation should be consistent; inconsistency will destroy the readers' confidence in the documentation.
- Third the documentation has to be pitched at the right level for its intended audience. A training manual cannot demand as much from its readers as a design document. We should use an appropriate level of formality and an appropriate vocabulary in our presentation.
- Finally we need to select the right approach in presenting information so that it will be readily understood. This certainly includes a choice of formal text, technical manual, workbook, tutorial, online help package and so on. Illustrations and tables are very important since they organize and present large amount of material for quick reference.

### **4.3 QUALITY ASSURANCE**

#### **4.3.1 Generic Risks**

Risk is the probability that undesirable things will happen. They are a consequence of inadequate information. Testing these systems should involve informed, conscious risk management. Risk management considers the severity of each hazard and the probability that an accident will result from the hazard. For each hazard the outcome of the risk assessment process is a statement of acceptability. We can't do everything- we have to make compromises, but we don't want to take risks that are unacceptably high. Some of the types of risks that are associated with the projects are given below.

#### **4.3.2 Intolerable**

The system must be designed in such a way that either the hazard doesn't arrive or if it does arrive it will not result in an accident.

### **4.3.3 As Low As Reasonably Practical (ALARP)**

The system must be designed in such a manner that the probability of an accident arising because of the hazard is completely minimized subject to other consideration such as cost, delivery and so on.

### **4.3.4 Acceptable**

While the system should be designed so as to reduce the probability of any hazard arising they should not be at the expense of increased cost, delivery time and other non-functional system attribute.

Once the potential hazards and their costs have been identified the system should be formulated in such a manner that these hazards are unlikely to result in an accident. Several approaches may be used which include.

### **4.3.5 Hazard probability Reduction**

The system is designed in such a manner that the probability of hazard arising is reduced. This is actually achieved by ensuring that the hazard cannot arise as result of a single failure or operating error.

### **4.3.6 Accident Prevention**

The system is designed in such a way that hazards can be detected. Other systems can then be activated to remove the hazard before an accident is caused. This approach is widely used in control systems.

#### 4.4 SYSTEM SECURITY

Any system that is developed should be secured and protected against possible hazards. The software takes care to see that in the event of interruption due to power failure or voltage fluctuation the data in the files are not erased. Since the program created the records then and there, interruptions do not erase all the keyed in input data or the content of any file.

Every system must provide built-in features for security and integrity of data. Without safeguards against unauthorized access, fraud, embezzlements, fire and natural disasters, a system could be so vulnerable as to threaten the survival of the organization.

To do an adequate job security a system analyst must analyze the risks, exposure and costs and specify measures such as passwords and encryption to provide protection. In addition backup copies of software and recovery restart procedures must be available when needed.

The program is protected to ensure that they are not tempted with the copies of all files are made as back file. In the user area, only compiled object codes are made available so that program could be protected for unauthorized modification.

As referred to the technical innovations and procedures applied to both hardware and operating system to protect against deliverable or accidental damage from defined threat.

## CHAPTER 5

### SYSTEM MAINTENANCE

It is impossible to produce software that doesn't need to be changed. Over the lifetime of any system the original requirements will be modified to reflect the changing user and customer needs. The system environment may change as new hardware is introduced. Errors undiscovered during system testing may emerge as a result. Hence the system maintenance is a process of changing a system after it has been delivered and is in use currently. These changes may be simple changes to correct coding errors, design errors, or even significant enhancement to correct specification errors to accommodate new requirements. Maintenance is therefore evolution of the system in accordance with the changes in environment. It is a process of changing a system to maintain its ability to survive. There are three different types of maintenance with very blurred distinction between them.

#### **Corruptive Maintenance**

It is concerned with fixing reported errors in the software. Coding errors are usually relatively cheap to correct. Design errors are more expensive as they involve the rewriting of several program components. Requirement errors are the most expensive to repair due to extensive system redesign that is involved.

#### **Adaptive Maintenance**

This involves changing the system to some new environments such as different network platform or for use with a different operating system. The system functionality does not radically change.

## **Perfective Maintenance**

This involves implementing new functional or non-functional system requirements, software customer policy changes, business changes are changes in top management.

The maintenance process is usually triggered by a set of change request from the users, management or customers. The cost and impact of these changes are first assessed. If proposed changes are accepted a new release of the system is planned. This release usually involves elements of adaptive, corrective as well as perfective maintenance. These changes are implemented and validated and a new version of the system is released.

Rather than viewing maintenance as a separate process it should be normally considered as an iteration of the development process. New requirements must be formulated and validated, components of the system must be redesigned and implemented and part or all of the system should be tested.

## **CHAPTER 6**

### **CONCLUSION**

This project 'Enterprise Resource Planning' has been designed and developed as per the specification of "JVC". It has been developed in VISUAL STUDIO .NET and SQL SERVER 7.0. it is very user friendly and provides necessary information through queries, which are easily accessible.

The current system drives away the handles of the previous one and meets the needs and provides valuable information to the management. This software can be operated easily. Any one can easily modify or upgrade this project whenever requirements arise from the user for more effective operations.

The project developed at present is efficient and effective, because it is tested repeatedly in all cases and can be implemented successfully at the site of usage. Project is effective because it is developed with standard codes and also we followed good programming rules to develop.

### **SCOPE FOR FUTURE ENHANCEMENT**

In this project the enhancement that can be added are,

- The branch details of the company can also be added further.
- Many reports can also be added as per the requirement of the company in the future.
- Individual reports can also be generated in future.

## APPENDIX 1

### TABLES

**Table A1.1 Admin Table**

<b>FIELD NAME</b>	<b>FIELD TYPE</b>	<b>WIDTH</b>	<b>DESCRIPTION</b>
Userid(PK)	Varchar	30	User identification
Pswd	Varchar	25	Password
Hintq	Varchar	80	Hint question
Hinta	Varchar	100	Hint answer
Cpswd	Varchar	25	Confirm password
Designation	Varchar	50	Designation
Department	Varchar	50	Department

**Table A1.2 Marketing Table**

<b>FIELD NAME</b>	<b>FIELD TYPE</b>	<b>WIDTH</b>	<b>DESCRIPTION</b>
Proposalid (PK)	Varchar	15	Proposal identification
Proposalname	Varchar	100	Proposal name
Proposaldate	DateTime	8	Proposal date
Deliverydate	DateTime	8	Delivery date
Noofproducts	Integer	8	Number of products
Soundtype	Varchar	35	Sound type
Screentype	Varchar	35	Screen type
Remotetype	Varchar	35	Remote type
Powertype	Varchar	35	Power type
Description	Varchar	250	Description of the product

**Table A1.3 Research and Development Table**

<b>FIELD NAME</b>	<b>FIELD TYPE</b>	<b>WIDTH</b>	<b>DESCRIPTION</b>
Rndid (PK)	Varchar	25	R & D Identification
Proposalid	Varchar	25	Proposal identification
Prodname	Varchar	65	Proposal name
Rnddate	DateTime	8	R & D Date
Reqqquantity	Varchar	15	Required quantity
Avaquantity	Varchar	15	Available quantity
Soundinfo	Varchar	50	Sound information
Screeninfo	Varchar	50	Screen information
Remoteinfo	Varchar	50	Remote information
Powerinfo	Varchar	50	Power information
Rnddescription	Varchar	350	R & D Description
Manpower	Varchar	5	Man power
Duration	Varchar	15	Duration

**Table A1.4 Warehouse Table**

<b>FIELD NAME</b>	<b>FIELD TYPE</b>	<b>WIDTH</b>	<b>DESCRIPTION</b>
Reqid (PK)	Varchar	25	Required Identification
Prodid	Varchar	25	Product identification
Prodname	Varchar	25	Product name
Requirement	Varchar	15	Requirement
Quantity	Varchar	15	Quantity

**Table A1.5 Purchase Order Table**

FIELD NAME	FIELD TYPE	WIDTH	DESCRIPTION
Purchaseid (PK)	Varchar	35	Purchase Identification
Requirement	Varchar	50	Requirement
Quantity	Varchar	15	Purchase quantity
Purchase_Date	DateTime	8	Purchase date

**Table A1.6 Quality Table**

FIELD NAME	FIELD TYPE	WIDTH	DESCRIPTION
Qualityid (PK)	Varchar	35	Quality Identification
Rndid (FK)	Varchar	35	R & D Identification
Soundqlty	Varchar	15	Sound quality
Screenqlty	Varchar	15	Screen quality
Remotqlty	Varchar	15	Remote quality
Powerqlty	Varchar	15	Power quality
Qty_Description	Varchar	350	Description of quality
Accept	Varchar	10	Acceptance

**Table A1.7 Manufacture Table**

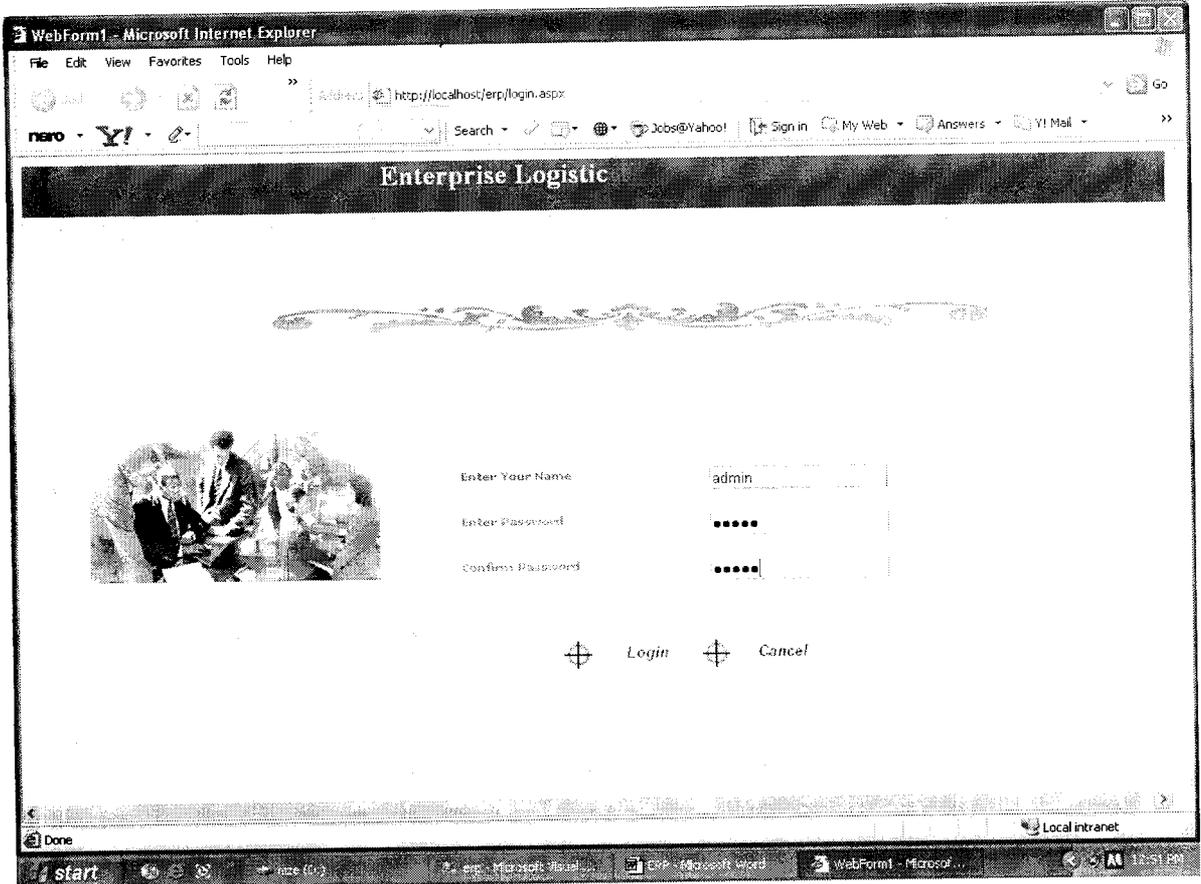
FIELD NAME	FIELD TYPE	WIDTH	DESCRIPTION
Manufactureid (PK)	Varchar	35	Manufacture identification
Rndid (FK)	Varchar	35	R & D Identification
Manufacturedate	DateTime	8	Manufacture date
Noofproducts	Varchar	12	Number of products manufactured

**Table A1.8 Sales Table**

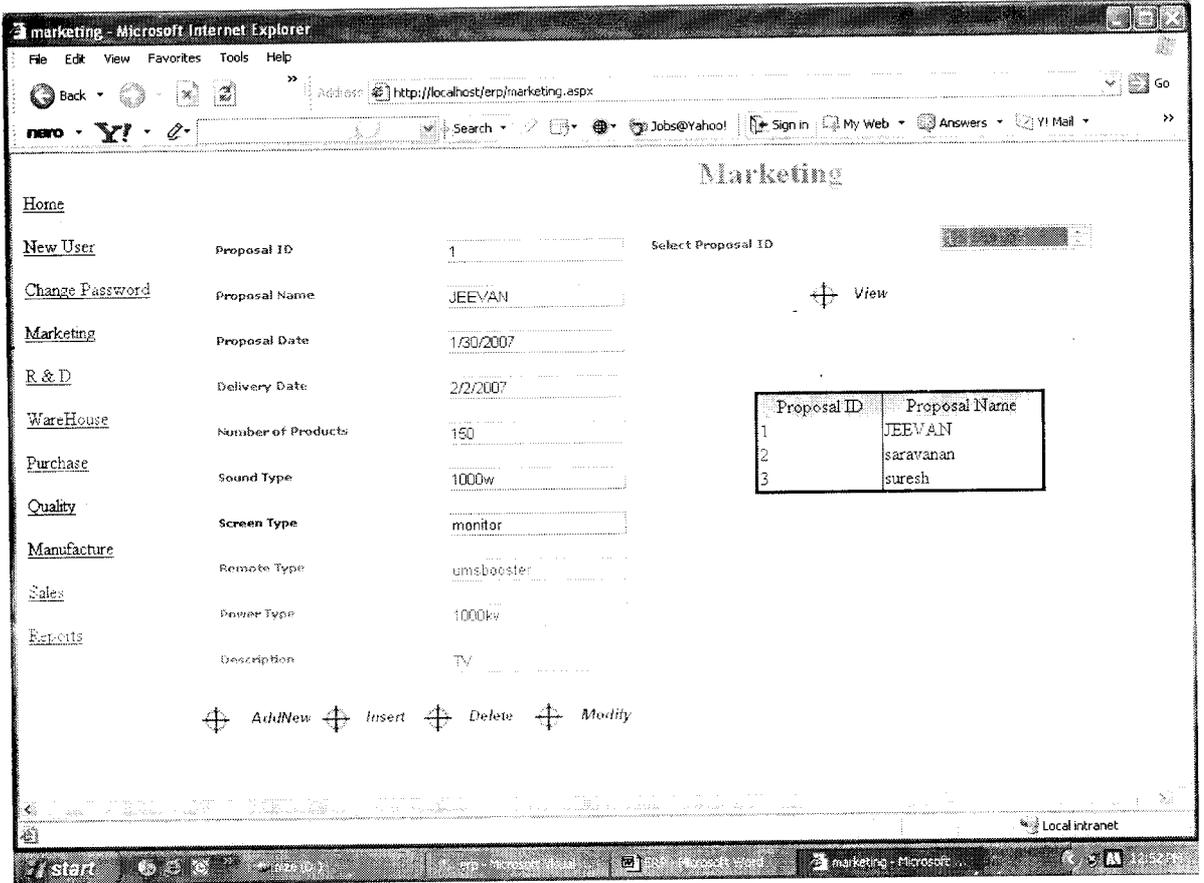
<b>FIELD NAME</b>	<b>FIELD TYPE</b>	<b>WIDTH</b>	<b>DESCRIPTION</b>
Sales_Id (PK)	Varchar	35	Sales Identification
Manufacture_Id (FK)	Varchar	35	Manufacture Identification
Sales_Date	DateTime	8	Sales date
Sale_nofprod	Varchar	15	Number of sales products
Proposal_nofprod	Varchar	15	Number of proposed products
Proposalid	Varchar	15	Proposal identification
Proposalname	Varchar	50	Proposal name
Proposaldate	DateTime	8	Proposal date
Deliverydate	DateTime	8	Delivery date

## Appendix

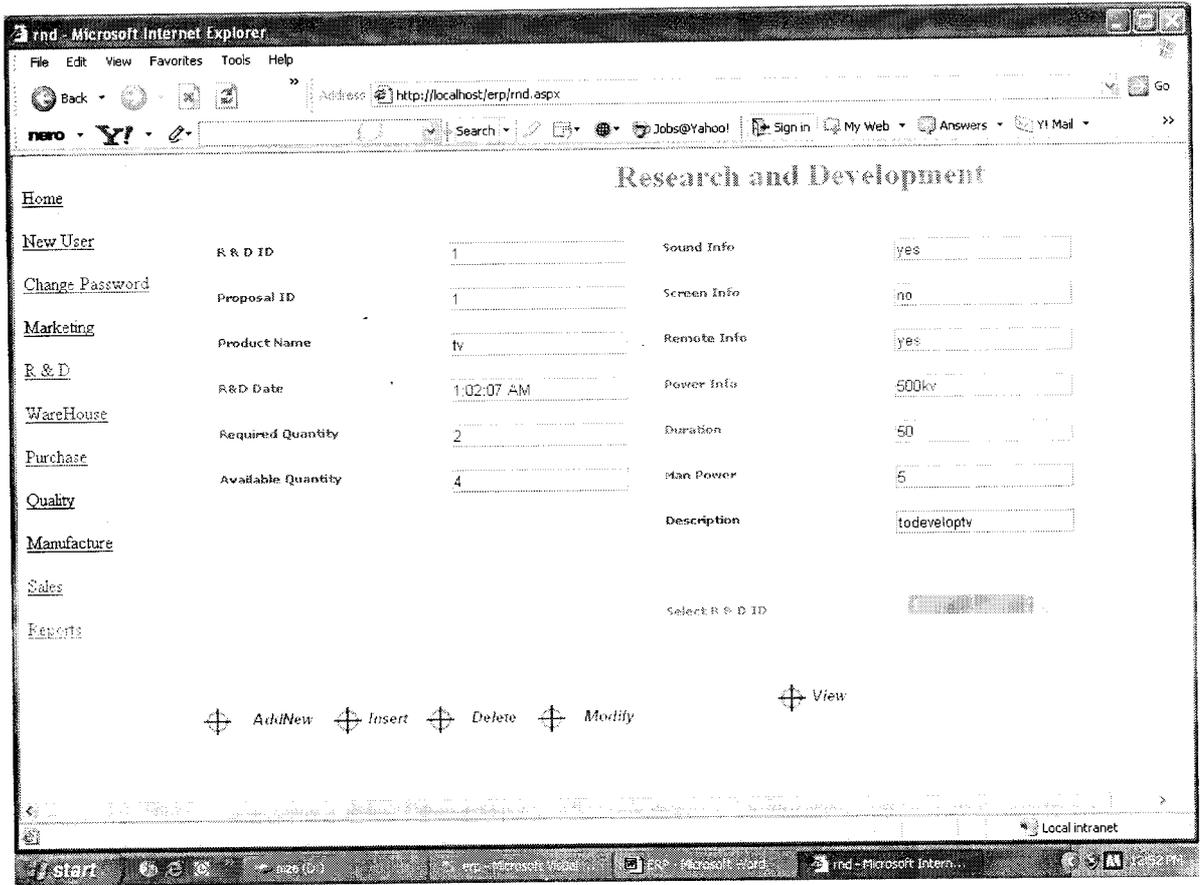
### A2 Screen Layout



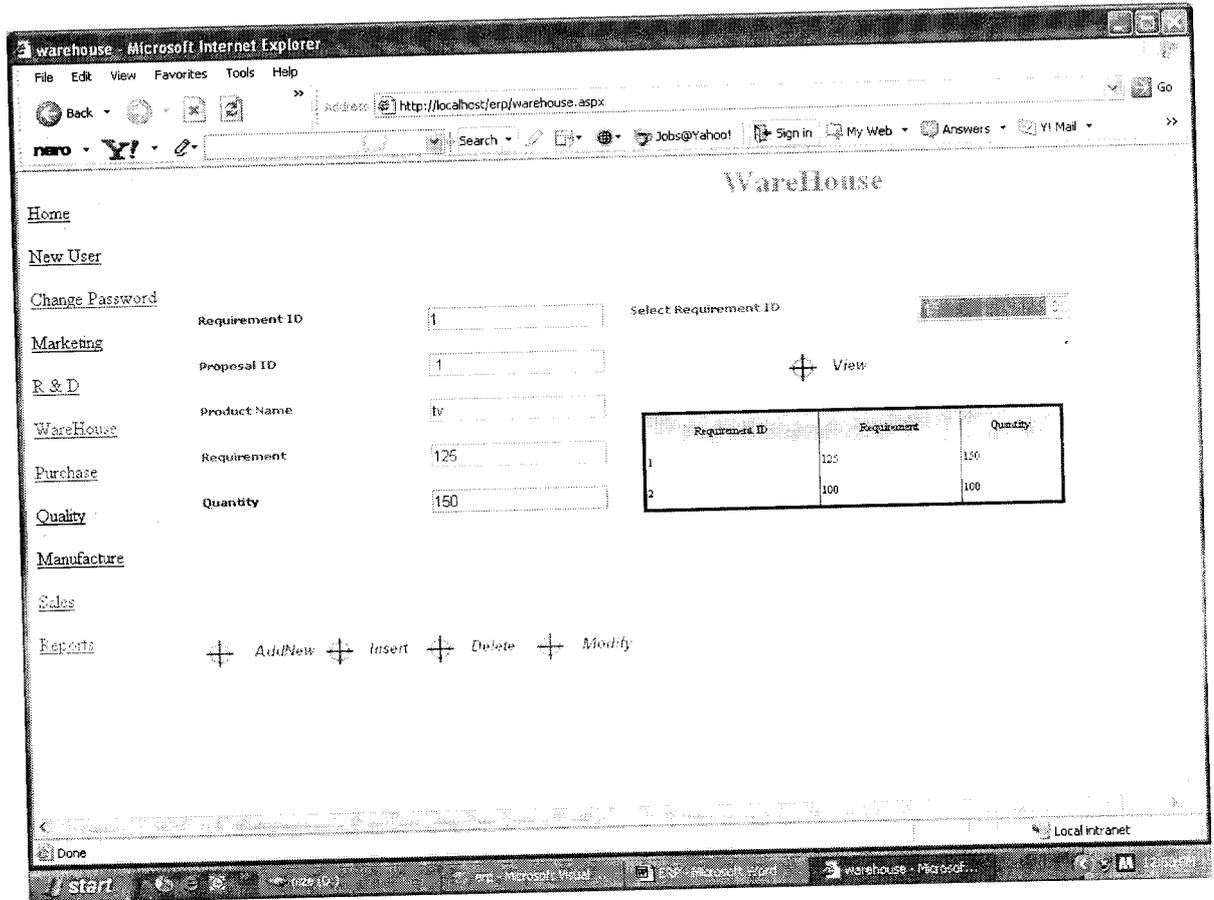
### LOGIN SCREEN



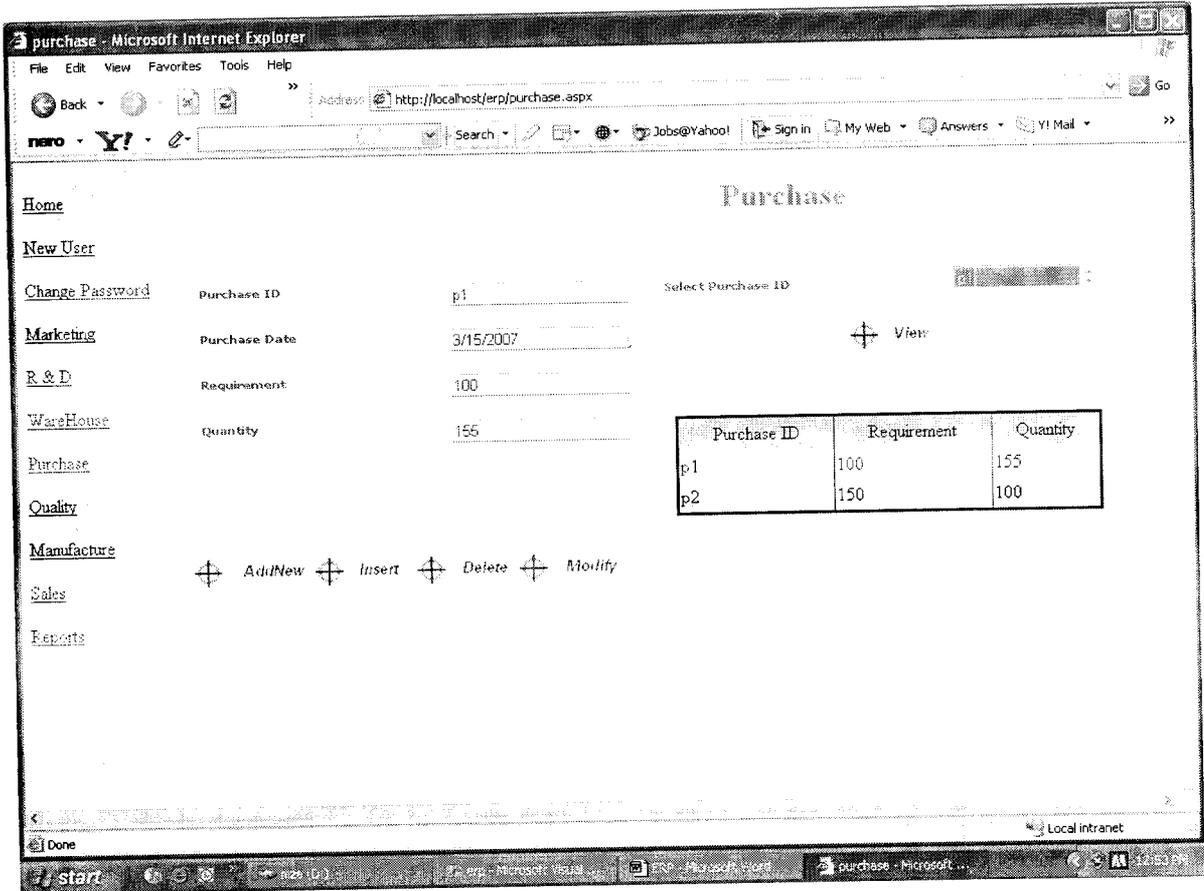
**MARKETING MODULE**



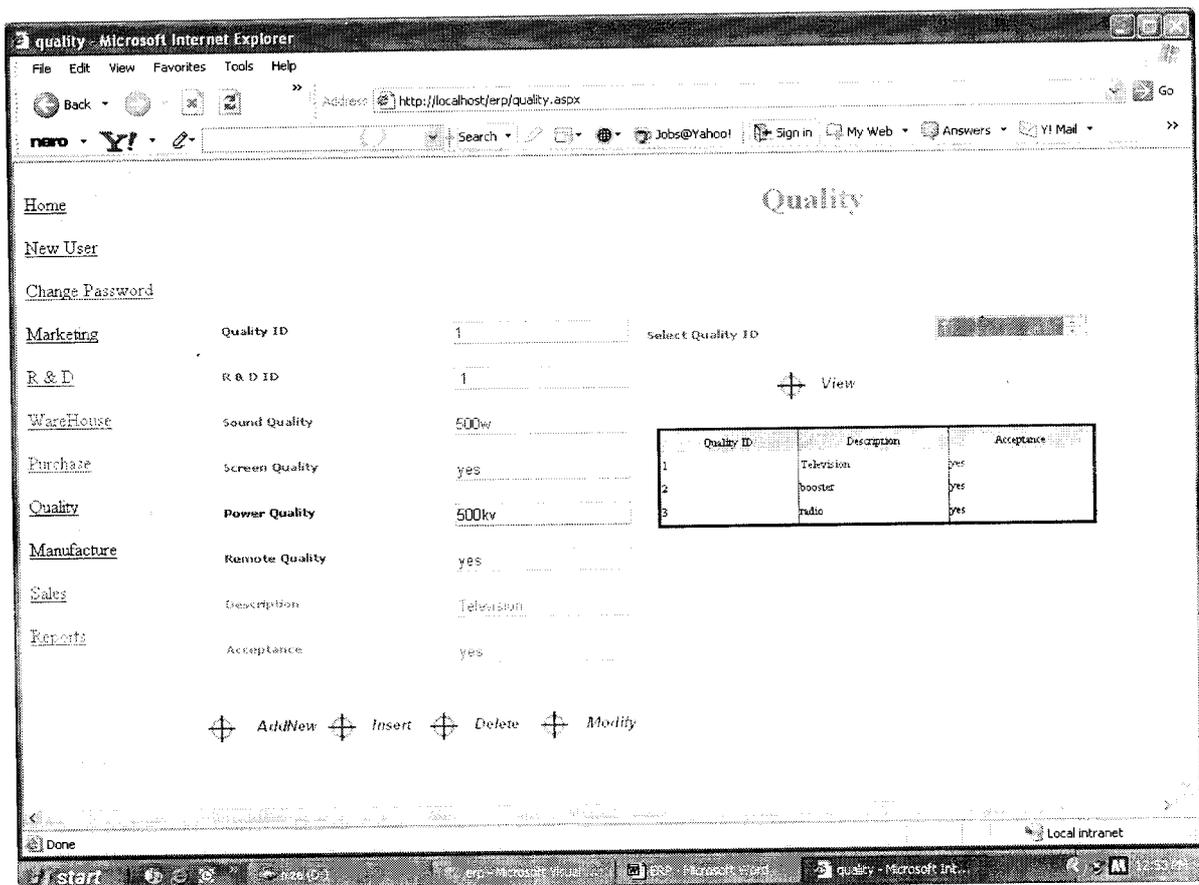
**RESEARCH AND DEVELOPMENT**



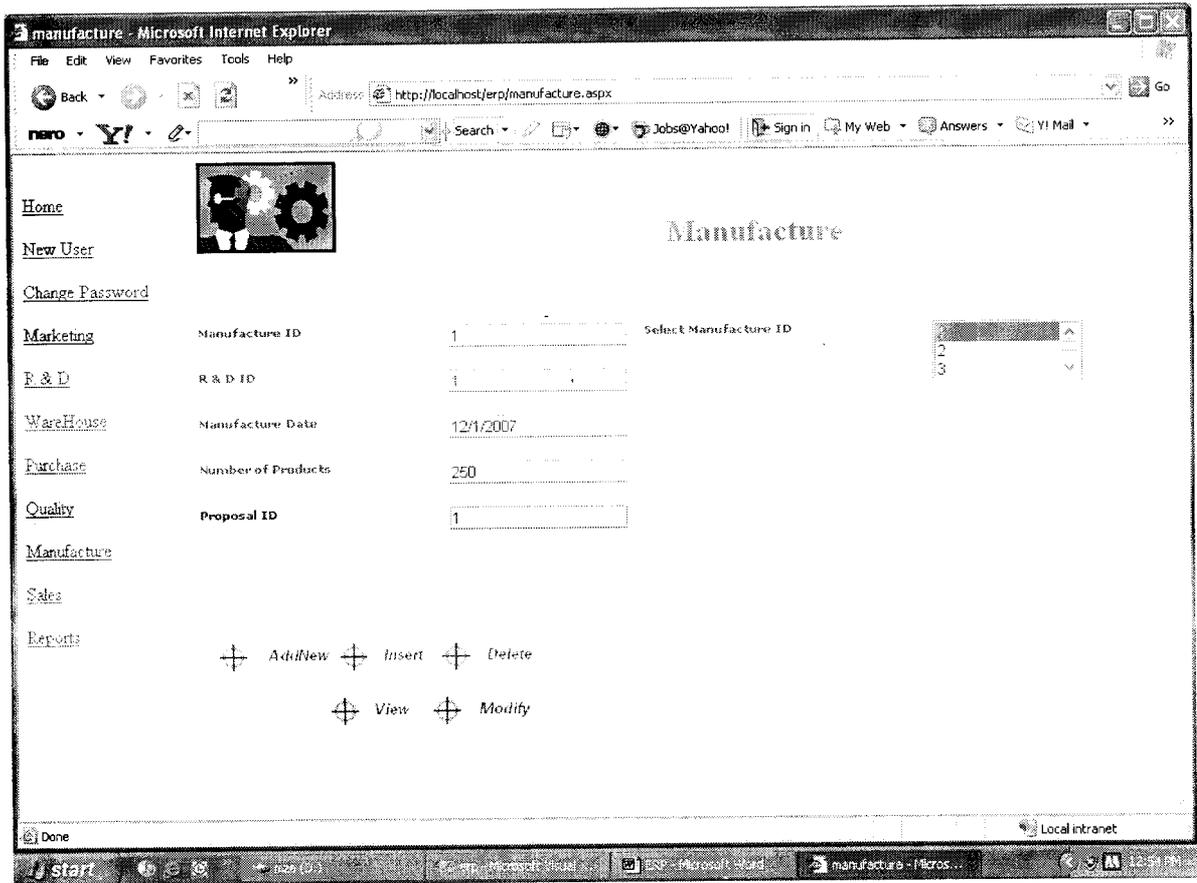
## WAREHOUSE DEPARTMENT



**PURCHASE DEPARTMENT**



**QUALITY DEPARTMENT**



## MANUFACTURE DEPARTMENT

sales - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Address http://localhost/erp/sales.aspx

Back Forward Stop Reload Home

Search Jobs@Yahoo! Sign in My Web Answers Y! Mail

## Sales

[Home](#)

[New User](#)      Sales ID:       Select Sales ID:

[Change Password](#)      Manufacture ID:      

[Marketing](#)      Proposal ID:

[R & D](#)      Sales Date:

[WareHouse](#)      Sales Quantity:

[Purchase](#)      Proposal Quantity:

[Quality](#)      Proposal Name:

[Manufacture](#)      Proposal Date:

[Sales](#)      Delivery Date:

[Rec-cts](#)

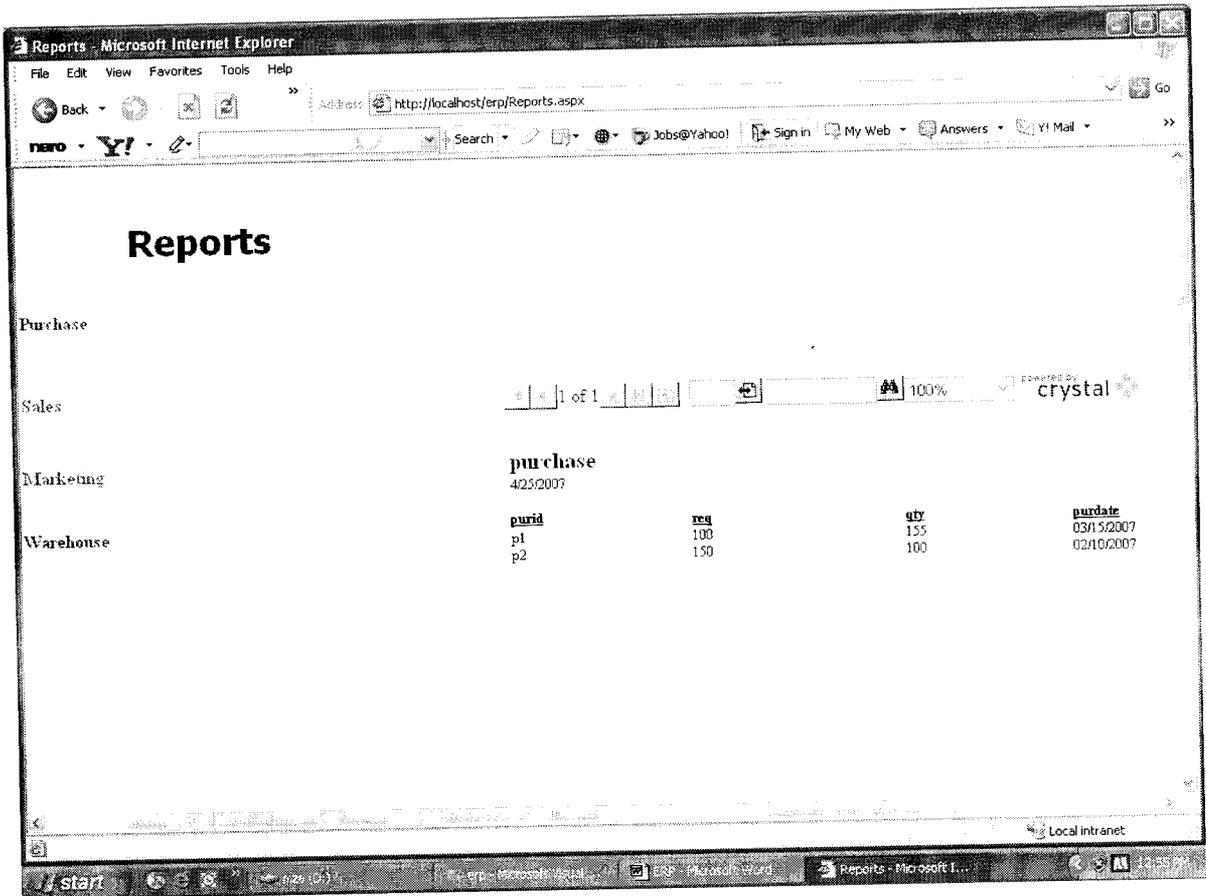
Sale ID	Sales Date	Sales Quantity	Pr
1	3/20/2007 12:00:00 AM	1000	1000

Local intranet

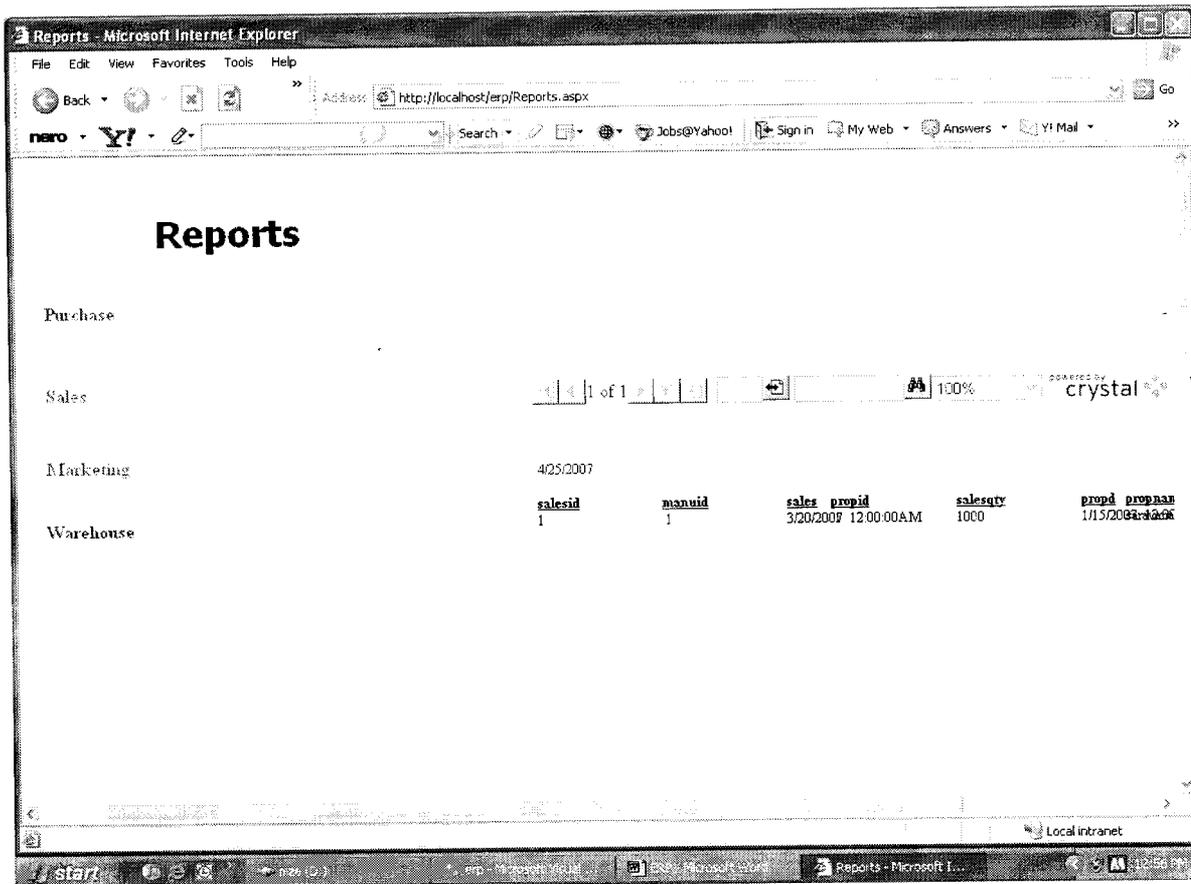
Done

start | size (0) | erp - Microsoft Visua | Microsoft Word | sales - Microsoft Inte... | M |

### SALES DEPARTMENT



**REPORTS FOR PURCHASE**



## REPORTS FOR SALES

**Reports**

Purchase

Sales

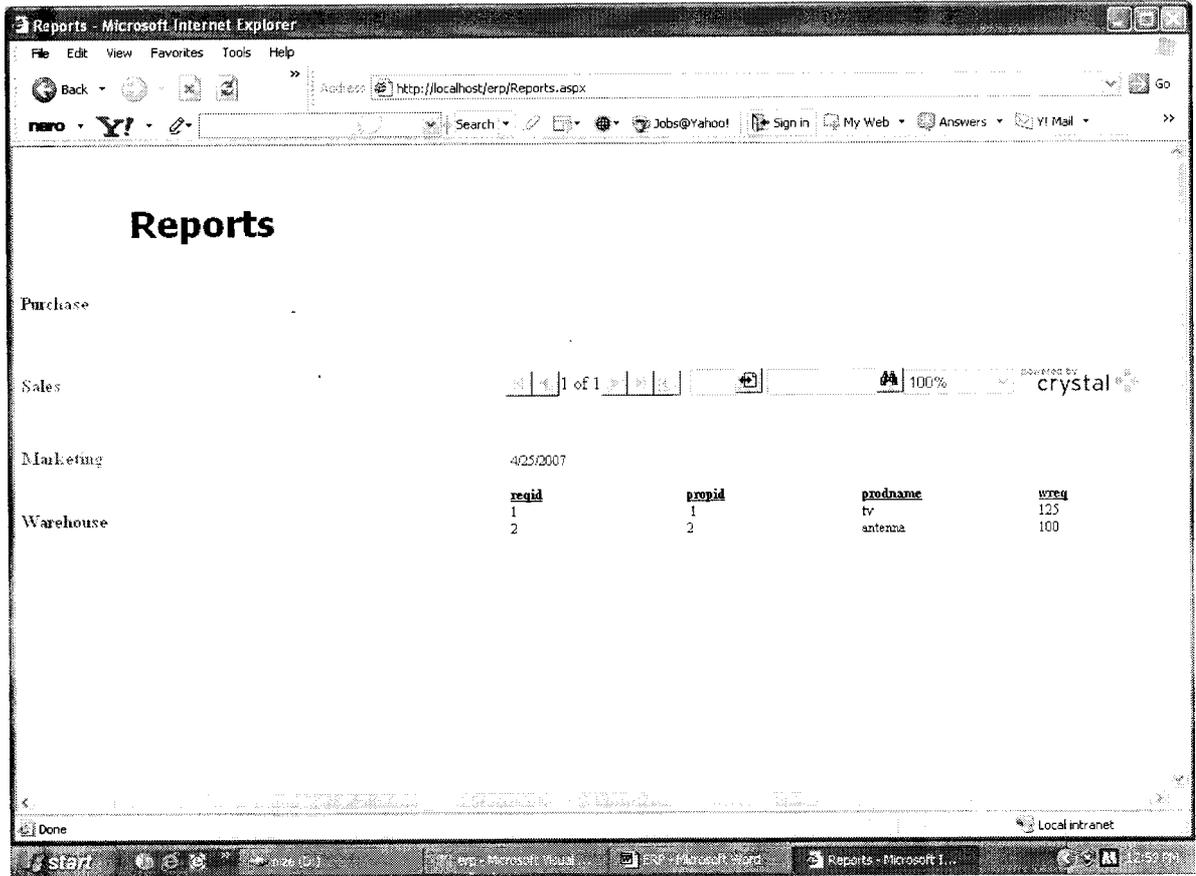
Marketing

Warehouse

Marketing Details  
4/25/2007

<u>propid</u>	<u>propname</u>	<u>propmat</u>	<u>delivdate</u>	<u>noprod</u>	<u>sdtype</u>	<u>scstype</u>	<u>remtype</u>
1	JEEVAN	01/30/2007	02/02/2007	150	1000w	monitor	unsbooster
2	sarwanan	03/15/2007	03/20/2007	100	500w	power	no
3	suresh	03/12/2007	03/02/2007	50	1000w	no	yes

**REPORTS FOR MARKETING**



**REPORTS FOR WAREHOUSE**

## REFERENCES

1. Andy Wigley, Stephen Wheelwright (2003) ' Microsoft .Net Compact Framework ' , Tata McGraw Hill, New Delhi.
2. Cars ten Thomsen (2001) 'Database Programming with Visual Basic .Net ' , Après, New York.
3. Damien Watkins ,Mark Hammond ,Brad Abrams (2003) ' Programming in the .Net Environment ' Pearson Education Singapore
4. Deitel H.M, Deitel P.J, Nieto T.R (2004) ' Visual Basic .Net How To Program' Second Edition, Pearson Education, Singapore.
5. Greg Buczek (2002) ' ASP.Net Tips And Techniques ' Tata McGraw Hill, New Delhi.
6. Matthew MacDonald (2002) 'The Complete Reference ASP.Net ' , Tata McGraw Hill, New Delhi.