SHOWROOM MANAGEMENT AND ANALYSIS SYSTEM

By

P. RAMESH

Reg. No: 71202621035

Of

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641006

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

June, 2005

BONAFIDE CERTIFICATE

Certified that this project report titled

SHOWROOM MANAGEMENT AND ANALYSIS SYSTEM

Is Bonafide work of

Mr. P.RAMESH (Reg. No: 71202621035)

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

The Candidate with University Register No. 71202621035 was examined by us in the project Viva-Voce examination held on 23-06-2025

Internal Examiner

External Examiner 23/6/25

ABSTRACT

Marketing is the major cause of success in every field. There is always a need for an organization or a company to get in touch with the people to market their product. Marketing a product is done in many ways. It may through advertisements, giving free samples to people, opening retail outlets, etc., Recently retail outlets has become a more success in the marketing industry. While following such marketing strategy, one should face many problems like maintaining the reorder level, handling the invoice in an efficient way, customer satisfaction by providing faster delivery, prediction of sales, etc.,

The Showroom Management and Analysis System (SMAS) is a solution to all the problems which we described earlier. With Showroom Management and Analysis System (SMAS) one handle purchase order processing, billing, stock levels, receipts and payments and also can generate various kinds of reports. The SMAS is fully a user friendly application. The Sales details of customer can be seen with a touch of a button. The Sales prediction will be help full in maintaining the stock level and handling the reorder level. Facilities movement analysis by identifying fast moving, slow moving item and non moving item which reduces inventory levels through better management of inventory and reduces holding cost and thus ensure optimum use of working capital.

The Showroom Management and Analysis System have the facilities easy access to data and past records for future analysis, reference and planning and reduce errors and subsequent substantial loss for the organization. The SMAS Eventually develops paperless outlet and a streamlined improved administration process. The Showroom Management and Analysis System have power to expend its function in future. The over all benefit of SMAS to TNPL is to enhance its image of establishment.

ACKNOWLEDGEMENT

At this pleasing moment of having successfully completed the project work, I wish to acknowledge my sincere gratitude and heartfelt thanks to our beloved Principal **Dr Dr.K.K.Padmanabhan B.Sc., (Engg), M.Tech, Ph.D.,** for giving me the adequate support and opportunity for completing this project work successfully.

I express my sincere thanks to **S.Thangasamy B.E (Honr), Ph.D**, the ever active and sympathetic, Head of the Department of Computer science & Engineering, who with his careful supervision has ensured me in attaining perfection of work.

I extend my sincere thanks to **Mr. A. MuthuKumar MCA M.phil,** Project Coordinator for rendering us all the timely helps through out the project.

I admit my heart full thanks to my guide **Mr. S. Hammed Ibrahim MCA.,** Lecturer, Department of Computer Science and Engineering, for his supervision, tremendous patience, active involvement and being supportive throughout the tenure of my project.

I also extend my gratitude to **Mr. K. Swaminathan, C.A** Chief Accounts Officer, Tamil Nadu Newsprint and Papers Limited for allowing me to carry out this project at their organization and guiding me in completing the project successfully.

Last but not the least I am grateful to my parents and friends for their constant support and encouragement.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 SHOWROOM MANAGEMENT AND ANALYSIS SYSTEM

The Showroom Management and Analysis System (SMAS) is versatile Software which automates the outlet of TNPL. The Showroom Management and Analysis System (SMAS) which handle invoice, purchase order processing, purchase inward entry with tax. discount, batch number and expiry date etc. The Billing process work is reduced and several billing types are introduced. The SMAS helps to analyze the stock, sales and produce a report which helps the outlet management to make a better decision.

Purpose:

The main objective of Showroom Management and Analysis System (SMAS) is to develop TNPL outlet a paperless outlet and increase customer satisfaction.

Benefits:

- Eventually develops paperless outlet.
- Streamlined and improved administration process.

- Enhance image of establishment.
- Adaptable for future expansion.
- Gives itemized billing containing batch No, quantity, rate, and amount.
- Facilities movement analysis by identifying fast moving, slow moving item and non moving item which reduces inventory levels through better management of inventory and reduces holding cost and thus ensure optimum use of working capital.
- Facilities easy access to data and past records for future analysis, reference and planning.
- Reduces errors and subsequent substantial loss.
- More customers can service daily.

1.2 ORGANIZATION PROFILE

- Promoted by the Government of Tamil Nadu in 1979
- Loan assistance extended by World Bank twice.
- Largest producer of Bagasse based paper in the world. Annual consumption of bagasse slated to touch 1 million tones per annum.
- Installed Capacity - 2, 30,000 tpa

Vision

To be the market leader in the manufacture of world class eco-friendly papers, by adopting innovative technologies for sustainable development. Down the ages, the art of paper making has seen tremendous technological innovations. A variety of materials have gone into paper making, initially towards perfection and later to enable mass production. Wood has emerged the favorite much to our collective chagrin. While every effort is now being made to stop the devastation of natural forests and rely on plantation wood to feed paper mills, the insatiable appetite for wood always poses a lurking threat to our priceless legacy the natural forests.

TNPL came up with an alternate technology. The idea was to make paper and newsprint primarily from Bagasse, namely sugarcane waste using as little wood as possible. TNPL's vision was to use renewable raw materials with minimum impact on the environment. So, TNPL pioneered the concept of Newsprint from bagasse. Wood use was reduced drastically. All the technological advantages of high-speed papermaking were brought in. TNPL today is a company with an installed capacity of 2, 30,000 tpa of Printing & Writing paper and Newsprint using primarily a renewable raw material, TNPL stands as the company, that has redefined the global perception of bagasse-based paper making. Come explore the fascinating world of TNPL - A Company which saw the future two decades back.

Technology

TNPL is acknowledged as the world leader in technology for the manufacture of newsprint from bagasse. TNPL has the most modern paper mill in the country with an unique bagasse procurement, storing, preserving, handling, processing and pulping system. The key areas of operation have been innovatively designed and developed by a superb technical team.

TNPL has two high-speed paper machines from Beloit Walmsley and Voith Paper. They have the unique flexibility to produce Newsprint as well as printing and writing paper. Today TNPL uses the Voith machine as a swing machine producing newsprint and printing and writing paper. TNPL derives its strength from its dedicated in-house R&D team, one of the best in the country. TNPL's customer-focused R&D has been instrumental in improving quality, ensuring consistency and introducing new products based on market feedback and needs.

Future

Providing viable solutions to future generations.TNPL, steadfast in its vision reaffirms its commitment to use renewable raw material. It has chosen a tough path in papermaking. And despite stiff competition from wood based Paper mills, TNPL has kept going and achieved more than anyone dreamed possible. Now, the focus is on consolidation, efficiency of its operations and an expanded product range. From achieving 100% self-sufficiency in power to introducing advanced computerized management systems; TNPL has excelled as the company of the future.

TNPL's eco-friendly paper, for a better tomorrow

TNPL has entered the new millennium with pride, the pride of having revolutionized the manufacture of paper with Bagasse. TNPL products have been well appreciated by both Indian and international communities.

Environment

TNPL is an unfolding saga of commitment to sustainable development. TNPL has carefully formulated an Environment policy. This policy keeps the future firmly in mind. Guided by principles of environmental responsibility, TNPL manufactures and markets papers using materials and processes that help minimize waste, conserve resources and protect our Earth's bio diversity. Its unfailing commitment to clean production techniques with minimum pollution load stands out as a bench mark in the industry.

As part of its commitment to the protection of the environment TNPL has installed 21.75 MW of capacity in its wind farms in south Tamil Nadu. TNPL's plan is to target an internal net zero carbon trade, provide green energy to the state grid using renewable resources to the extent it has to rely on fossil fuels. TNPL is also increasingly relying on for its fuel requirement.

Recognizing TNPL's Commitment to preserving nature, the World Wide Fund for Nature - India (WWF) has endorsed TNPL as an Eco-conscious company and has franchised its PANDA Logo to be used on TNPL's branded products.

TNPL, in February 2002 has become an ISO14001 company. This amply demonstrates its commitment to protect the environment.

CHAPTER 2

REQUIREMENT SPECIFICATION

An SRS is basically an organization's understanding (in writing) of a customer or potential client's system requirements and dependencies *at a particular point in time* (usually) prior to any actual design or development work. It's a two-way insurance policy that assures that both the client and the organization understand the other's requirements from that perspective at a given point in time.

It's important to note that an SRS contains functional and nonfunctional requirements only; it doesn't offer design suggestions, possible solutions to technology or business issues, or any other information other than what the development team understands the customer's system requirements to be.

A well-designed, well-written SRS accomplishes four major goals:

- It provides feedback to the customer.
- It decomposes the problem into component parts.
- It serves as an input to the design specification.
- It serves as a product validation check.

SRSs are typically developed during the first stages of "Requirements Development," which is the initial product development phase in which information is gathered about what requirements are needed--and not. This information-gathering stage can include onsite visits, questionnaires, surveys, interviews, and perhaps a return-on-investment (ROI) analysis or needs analysis of the customer or client's current business environment. The actual specification, then, is written after the requirements have been gathered and analyzed.

2.1 HARDWARE REQUIREMENTS

• Processor	:	Intel Pentium
• Processor	:	500 MHZ
• Memory	:	96 MB RAM
• Hard Disk	:	10 GB

2.2 SOFTWARE REQUIREMENTS

• Operating System	:	Windows 2000/XP/NT Server
• Programming Language	:	VB.NET
• Database Server	:	Microsoft SQL Server 2000

2.3 SOFTWARE OVERVIEW

2.3.1 What Is Microsoft .NET?

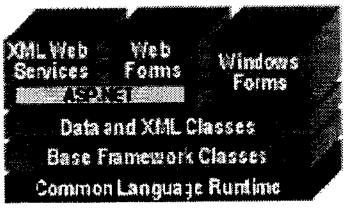
Microsoft .NET is software that connects information, people, systems, and devices. It spans clients, servers, and developer tools, and consists of:

- The .NET Framework used for building and running all kinds of software, including Web-based applications, smart client applications, and XML Web services—components that facilitate integration by sharing data and functionality over a network through standard, platform-independent protocols such as XML (Extensible Markup Language), SOAP, and HTTP.
- Developer tools, such as Microsoft Visual Studio® .NET 2003 which provides an integrated development environment (IDE) for maximizing developer productivity with the .NET Framework.
- A set of servers, including Microsoft Windows® Server 2003, Microsoft SQL Server™, and Microsoft BizTalk® Server, that integrates, runs, operates, and manages Web services and Web-based applications.

- Client software, such as Windows XP, Windows CE, and Microsoft Office XP, that helps developers deliver a deep and compelling user experience across a family of devices and existing products.

2.3.2 What is the .NET Framework?

The .NET Framework is a development and execution environment that allows different programming languages & libraries to work together seamlessly to create Windows-based applications that are easier to build, manage, deploy, and integrate with other networked systems.



The .NET Framework consists of:

- **The Common Language Runtime (CLR)**

The common language runtime (CLR) is responsible for run-time services such as language integration, security enforcement, and memory, process, and thread management. In addition, the CLR has a role at development time when features such as life-cycle management, strong type naming, cross-language exception handling, and dynamic binding reduce the amount of code that a developer must write to turn business logic into a reusable component.

- **The Framework Class Libraries (FCL)**

A consistent, object-oriented library of prepackaged functionality Base classes provide standard functionality such as input/output, string manipulation, security management, network communications, thread management, text management, and user interface design features.

The ADO.NET classes enable developers to interact with data accessed in the form of XML through the OLE DB, ODBC, Oracle, and SQL Server interfaces. XML classes enable XML manipulation, searching, and translations. The ASP.NET classes support the development of Web-based applications and Web services. The Windows Forms classes support the development of desktop-based smart client applications.

Together, the class libraries provide a common, consistent development interface across all languages supported by the .NET Framework.

2.3.3 Reasons for Using VB.NET

- **Seamless Deployment**. VB .NET promises to end "DLL Hell" and it may. I haven't experimented with it much yet but I've heard mixed reviews. Probably it will be better once we're used to it. It can't be a whole lot worse.
- **More Robust Code**. The features they list here are the real-time background compiler (to check syntax as you type) and the task list. Neither of these is a big step beyond VB 6. They also list strict type checking (which is good) and structured error handling. Structured error handling is really just more like the way it works in C/C++. It has some advantages but also some big disadvantages such as no Resume statements and no On Error Resume Next capabilities.
- **Powerful Windows-base Applications**. This isn't a real big deal. They point to the Form Designer (we've always had that), control anchoring and docking (replaces straightforward code with confusing design time properties), in-place menu editor (the old one was fine), and the tab order editor (a nice but tiny feature).

- **Powerful, Flexible Data Access**. They say VB .NET provides ADO and ADO.NET. Actually it looks like Microsoft is trying to dump ADO. ADO.NET does have some nice features, but it's aimed strongly at Web-like program structures where database users never interfere with each other. Record locking doesn't work as it does in ADO so building a traditional desktop application for multiple users requires some trickery.

- **Simplified Component Creation**. The Toolbox now includes more non-visual components (sort of like the Timer control) for things like database connection, Dataset, message queues, etc. It's not hard to create these objects in your code. Putting them on the form lets you do a bit more design-time configuration for them so this isn't a bad thing; it's just no big deal.

- **Enhanced Control Creation**. They don't say much here. It's true that VB .NET lets you build user controls. I haven't seen any big advantages over how it works in VB5/6 but I haven't spent a huge amount of time on this. One BIG disadvantage is the user control object (and the Picture Box and Form for that matter) no longer have an Auto Redraw property. If you want Auto Redraw, you need to implement it yourself with your own bitmap buffers.

- **Complete, Direct Access to the Platform**. This just means VB programmers can use the .NET Framework. That gives access to the registry, event log; performance counters, and files system. What this means in practice is you use new classes to do stuff that you would have done using API calls before. A nice feature but nothing you couldn't live without.

- **Integrated Reporting with Crystal Reports** (Professional edition and above). This lets you manipulate Crystal Reports within the IDE. I suppose you might care if you use Crystal Reports a lot.

2.3.4 Database Design

The OUTLET MANAGEMENT AND ANALYSIS SYSTEM uses Microsoft SQL Server 2000 as its Backend. Microsoft Extends the Performance, reliability, quality and ease- of-use of Microsoft SQL Server version 7.0. Microsoft SQL Server 2000 includes several new features that make it an excellent database platform for large-scale online transactional processing (OLTP), data warehousing, and e-commerce Application.

2.3.5 Fundamentals of SQL Server 2000 Architecture

- Online Transaction Processing (OLTP) systems must be capable of handling thousands of orders placed at the same time.

- Increasing numbers of corporations are implementing large Web sites as a mechanism for their customers to enter orders, contact the service department, get information about products, and for many other tasks that previously required contact with employees. These sites require data storage that is secure, yet tightly integrated with the Web.

- Organizations are implementing off-the-shelf software packages for critical services such as human resources planning, manufacturing resources planning, and inventory control. These systems require databases capable of storing large amounts of data and supporting large numbers of users.

- Organizations have many users who must continue working when they do not have access to the network. Examples are mobile disconnected users, such as traveling sales representatives or regional inspectors. These users must synchronize the data on a notebook or laptop with the current data in the corporate system, disconnect from the network, record the results of their work while in the field, and then finally reconnect with the corporate network and merge the results of their fieldwork into the corporate data store.

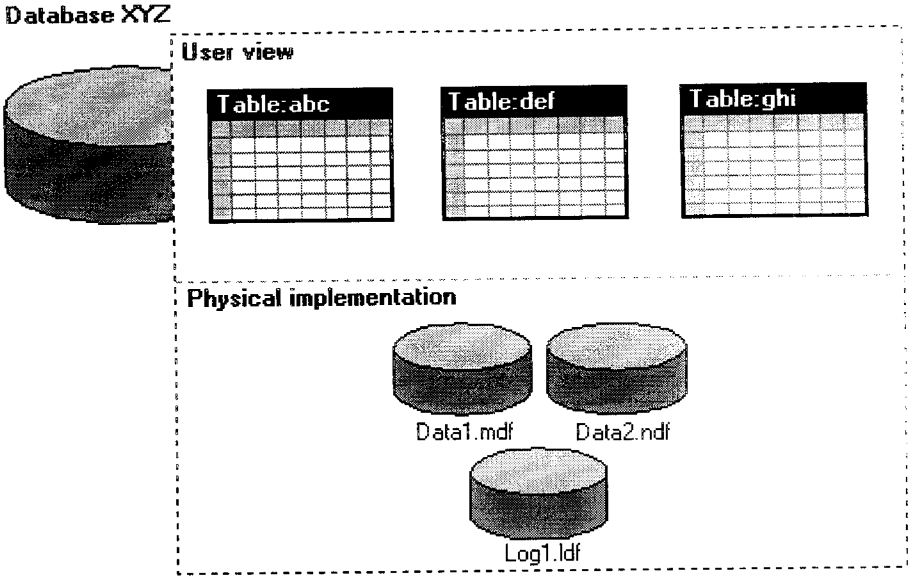
- Managers and marketing personnel need increasingly sophisticated analysis of trends recorded in corporate data. They need robust Online Analytical Processing (OLAP) systems easily built from OLTP data and support sophisticated data analysis.

- Independent Software Vendors (ISVs) must be able to distribute data storage capabilities with applications targeted at individuals or small workgroups. This means the data storage mechanism must be transparent to the users who purchase the application. This requires a data storage system that can be configured by the application and then tune itself automatically so that the users do not need to dedicate database administrators to constantly monitor and tune the application.

Database Architecture

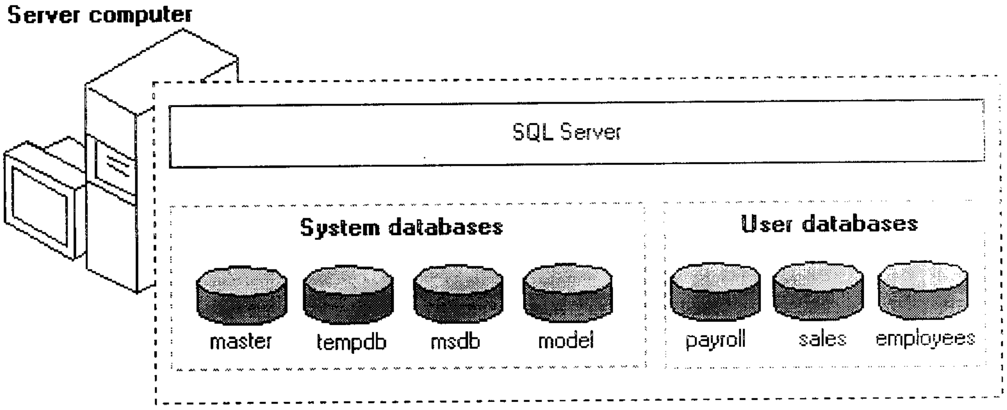
Microsoft® SQL Server™ 2000 data is stored in databases. The data in a database is organized into the logical components visible to users. A database is also physically implemented as two or more files on disk.

When using a database, you work primarily with the logical components such as tables, views, procedures, and users. The physical implementation of files is largely transparent. Typically, only the database administrator needs to work with the physical Implementation.



2.1 Database Architecture

Each instance of SQL Server has four system databases (**master**, **model**, **tempdb**, and **msdb**) and one or more user databases. Some organizations have only one user database, containing all the data for their organization. Some organizations have different databases for each group in their organization, and sometimes a database used by a single application. For example, an organization could have one database for sales, one for payroll, one for a document management application, and so on. Sometimes an application uses only one database; other applications may access several databases.



2,2 SQL Server

It is not necessary to run multiple copies of the SQL Server database engine to allow multiple users to access the databases on a server. An instance of the SQL Server Standard or Enterprise Edition is capable of handling thousands of users working in multiple databases at the same time. Each instance of SQL Server makes all databases in the instance available to all users that connect to the instance, subject to the defined security permissions.

When connecting to an instance of SQL Server, your connection is associated with a particular database on the server. This database is called the current database. You are usually connected to a database defined as your default database by the system administrator, although you can use connection options in the database APIs to specify another database. You can switch from one database to another using either the Transact-SQL USE database name statement, or an API function that changes your current database context.

SQL Server 2000 allows you to detach databases from an instance of SQL Server, then reattach them to another instance, or even attach the database back to the same instance. If you have a SQL Server database file, you can tell SQL Server when you connect to attach that database file with a specific database name.

2.3.6 FEATURES OF SQL SERVER 2000

- Internet Integration.

The SQL Server 2000 database engine includes integrated XML support. It also has the scalability, availability, and security features required to operate as the data storage component of the largest Web sites. The SQL Server 2000 programming model is integrated with the Windows DNA architecture for developing Web applications, and SQL Server 2000 supports features such as English Query and the Microsoft Search Service to incorporate user-friendly queries and powerful search capabilities in Web applications.

- Scalability and Availability.

The same database engine can be used across platforms ranging from laptop computers running Microsoft Windows® 98 through large, multiprocessor servers running Microsoft Windows 2000 Data Center Edition. SQL Server 2000 Enterprise Edition supports features such as federated servers, indexed views, and large memory support that allow it to scale to the performance levels required by the largest Web sites.

- Enterprise-Level Database Features.

The SQL Server 2000 relational database engine supports the features required to support demanding data processing environments. The database engine protects data integrity while minimizing the overhead of managing thousands of users concurrently modifying the database. SQL Server 2000 distributed queries allow you to reference data from multiple sources as if it were a part of a SQL Server 2000 database, while at the same time, the distributed transaction support protects the integrity of any updates of the distributed data. Replication allows you to also maintain multiple copies of data, while ensuring that the separate copies remain synchronized. You can replicate a set of data to multiple, mobile, disconnected users, have them work autonomously, and then merge their modifications back to the publisher.

- Ease of installation, deployment, and use.

SQL Server 2000 includes a set of administrative and development tools that improve upon the process of installing, deploying, managing, and using SQL Server across several sites. SQL Server 2000 also supports a standards-based programming model integrated with the Windows DNA, making the use of SQL Server databases and data warehouses a seamless part of building powerful and scalable systems. These features allow you to rapidly deliver SQL Server applications that customers can implement with a minimum of installation and administrative overhead.

- Data warehousing.

SQL Server 2000 includes tools for extracting and analyzing summary data for online analytical processing. SQL Server also includes tools for visually designing databases and analyzing data using English-based questions.

Tables can be created in two ways.

1. **SQL Query Analyzer.**
2. **SQL Enterprise Manager.**

Microsoft SQL Server 2000 SQL Query Analyzer is a graphical tool that allows you to:

- Create queries and other SQL scripts and execute them against SQL server databases.
- Quickly create commonly used database object from predefined scripts.
- Quickly copy existing database objects.
- Debug and Execute stored procedures.
- Debug query performance problems.
- Quickly insert, update, or delete rows in a table.(Open Table Window)

Microsoft SQL Server Enterprise Manager is the primary administrative tool for Microsoft SQL Server 2000 and provides a Microsoft management

Console (MMC)-Compliant user interface that allows user to:

- Define groups of servers running SQL Server.
- Register Individual servers in a group.
- Configure all SQL Server options for each registered server.
- Create and administer all Server databases, objects, logins, users and permissions in each registered server.

CHAPTER 3

SYSTEM ANALYSIS

3.1 PROBLEM DEFINITION

The traditional way of handling the outlet is now outdated due to advancement in field of information technology. The TNPL has planned to give its customer the benefits of modern outlet management.

3.2 SYSTEM ANALYSIS

3.2.1 EXISTING SYSTEM

The TNPL has planned to start its own outlet for its invaluable customer. They are planning to provide a fullest benefit of information technology in an outlet, which result in Showroom Management and Analysis System (SMAS).

DRAWBACKS OF THE TRADITIONAL SYSTEM:

- Time consuming and Laborious.
- Paper full environment.
- Chances for error and Inaccurate.
- Difficult to Prediction sales.
- Processing customer order involves time.
- Chanced to get out to stock.

3.2.2 PROPOSED SYSTEM

The Showroom Management and Analysis System (SMAS) is a solution to all the problems which we described earlier. With Showroom Management and Analysis System (SMAS) one handle purchase order processing, billing, stock levels, receipts and payments and also can generate various kinds of reports. It gives the user a more convenient

ADVANTAGES OF THE PROPOSED SYSTEM:

- Eventually develops paperless outlet.
- Streamlined and improved administration process.
- Enhance image of establishment.
- Gives itemized billing containing batch No, quantity, rate, and amount.
- Facilities movement analysis by identifying fast moving, slow moving item and non moving item which reduces inventory levels through better management of inventory and reduces holding cost and thus ensure optimum use of working capital.
- Facilities easy access to data and past records for future analysis, reference and planning.
- Reduces errors and subsequent substantial loss.
- More customers can service daily.

3.3 FEASIBILITY STUDY

All projects are feasible given unlimited resources and infinite time. It is both necessary and prudent to evaluate the feasibility of the project at the earliest possible time. Feasibility and risk analysis are related in many ways. If project risk is great, the feasibility listed below are equally important. The following feasibility techniques have been used in this project.

3.3.1 ECONOMIC FEASIBILITY

In Economic feasibility, we should check whether there are sufficient benefits in creating the system to make the cost acceptable or not. If the cost of the project is very high than its benefits, then it will not satisfy the economic feasibility. In this project, the benefit is well satisfied than the cost needed to develop it.

The Showroom Management and Analysis System is designed in such a way that the financial benefits exceeds or equals the cost. It has been decided to develop a system at reasonable costs. The system is sure to be good investment for the organization, since it requires minimal cost of implementing.

3.3.2 TECHNICAL FEASIBILITY

In Technical feasibility, we should check whether the project can be done with the current equipment and existing software technology. So that it is technically feasible. In this project we used the existing software technology. Going for a new software will be technically complex.

The proposed system is designed to operate under minimal technical requirements. There is a wide range of future enhancements that can be implemented in the system. The tools used are highly reliable and guarantee case of access, data security and accuracy. The proposal is technically feasible.

3.3.3 OPERATIONAL FEASIBILITY

In operational feasibility, we should check whether the software will be used after developed and implemented. The proposed system will be very useful for the organization since an efficient **SHOWROOM MANAGEMENT AND ANALYSYS SYSTEM** would contribute for reducing the expenses as well as acquiring good products from well-reputed companies. Thereby the proposed system is highly operable.

3.3.4 JUSTIFICATION

The project has the minimal software and hardware requirements making it technically feasible. It can also be implemented in the organization with the guarantee of satisfying the user needs. The project is highly operable. The above details specify that the project is feasible to all situations.

CHAPTER 4

SYSTEM DESIGN

4.1 MODULAR DESIGN

A Modular design reduces complexity, facilities change (a critical aspect of software maintainability), and results in easier implementation by encouraging parallel development of different parts of a system. Software with effective modularity is easier to develop because function may be compartmentalized and interfaces are simplified. Software architecture embodies modularity, that is, software is divided into separately named and addressable components called modules that are integrated to satisfy problem requirements. Modularity is the single attribute of software that allows a program to be intellectually manageable. Monolithic software (i.e., a large program comprised of a single module) cannot be easily grasped by reader. The five important criteria that enable us to evaluate a design method with respect to its ability to define an effective modular design are,

- **Modular Decomposability**

  If a design method provides a systematic mechanism for decomposing the problem into sub problems, it will reduce the complexity of the overall problem, thereby achieving an effective modular solution.

- **Modular Compensability**

  If a design method enables existing design components to be assembled into a new system, it will yield a modular solution that does not reinvent the wheel.

- **Modular Understandability**

If a module can be understood as a standalone unit, it will be easier to build and easier to change.

- **Modular Continuity**

If small changes to the system requirements result in changes to individual modules, rather than system-wide changes, the impact of change-induced side effects will be minimized.

- **Modular Protection**

If an aberrant condition occurs within a module and its effects are constrained within that module, the impact of error-induced side effects will be minimized.

4.2 INTERFACE DESIGN

Interface design mainly focuses on the design of interfaces between software modules, external entities and the user. The design of internal program interfaces, sometimes called inter modular interface design, is driven by the data that must flow between modules and the characteristics of the programming language in which the software is to be implemented External interface design begins with an evaluation of each external entity represented in the DFDs of the analysis model. Both internal and external interface design must be coupled with data validation and error handling algorithms within a module.

Because side effects propagate across program interfaces, it is essential to check all data flowing from module to module to ensure that the data conform to bounds established during requirements analysis. User interface design has as much to do with the study of people as it do with technology issues. Who is the user? How does the user learn to interact with a new computer-based system? So the system should be developed in a user-friendly manner.

4.3 FUNCTIONAL INDEPENDENCE

The concept of functional independence is a direct outgrowth of modularity and the concepts of abstraction and information hiding. It is achieved by developing modules with "single-minded" function and an "aversion" to excessive interaction with other modules. Independent modules are easier to maintain because secondary effects caused by the design/code modification are limited, error propagation is reduced, and reusable modules are possible. To summarize, functional independence is a key to good design.

4.4 COHESION

Cohesion is a natural extension of the information hiding concept. A cohesive module performs a single task within a software procedure, requiring little interaction with procedures being performed in other parts of a program. A module that performs a set of tasks that relate to each other loosely, if at all, is termed coincidentally cohesive. A module that performs tasks that are related logically is logically cohesive. When a module contains tasks that are related by the fact that all must be executed within the same span of time, the module exhibits temporal cohesion. Moderate levels of cohesion are relatively close to one another in the degree of module independence. When processing elements of a module are related and must be executed in a specific order, procedural cohesion exists. When all processing elements concentrate on one area of a data structures, communicational cohesion is present. High cohesion is characterized by a module that performs one distinct procedural task.

4.5 DATABASE DESIGN

Oracle is chosen for developing the database. For the optimum design of the database, to have a better response time and data integrity, avoid data redundancy, and for security of the database are some of the reasons behind the choice of Oracle.

4.5.1 NORMALIZATION

First Normal Form

It states that the domain of an attribute must include only atomic value and that the value of any attribute in a tuple must be a single value from the domain of that attribute. Hence 1NF disallows having a set of values or a combination of both as an attribute value for a single tuple .In other words 1NF disallows relations within relations or relations as attribute of tuples. The only attribute values permitted by 1NF are single atomic values. All the tables in the database satisfy the first normal form.

Second Normal Form

The test for 2NF involves testing for functional dependencies whose left hand side attribute are part of the primary key. If the primary key contains a single attribute the test need not be applied at all. A relation schema *R* is in 2NF if every non prime attribute *A in R is* functionally dependent on the primary key of R. All the tables with two attributes in the primary key field satisfy second normal form.

Third Normal Form

A functional dependency *X->Y* in a relation schema *R* is a transitive dependency if there is neither a set of attributes Z that is neither a candidate key nor a subset of any key of R and both X -> Z and Z -> Y hold.

4.6 I/P AND O/P DESIGN

4.6.1 INPUT DESIGN

Input Design is the process of converting the user originated inputs to a computer based format. The design decisions for handling input specify how data are accepted for computer processing. Input design is a part of overall system design that needs careful attention and it includes specifying the means by which actions are to be taken.

A system user interfacing through a workstation must be able to tell the system whether to accept input, produce a report, or end the processing. The collection of input data is regarded to be the most expensive part of the overall system design. Since the inputs have to be planned in such a manner so as to get the relevant information, extreme care is taken to obtain the pertinent information. If the data fed into the system is incorrect, then the processing as well as the outputs will magnify these errors .The arrangement of messages and comments in online conversations, as well as, the placement of data, headings, and titles on display screens or source documents, is also part of the input design. An improper design that leaves the screen blank will confuse a user about what action to be taken next. Online systems include a dialogue or conversation between the user and the system. The input design consists of developing specifications and procedures for data preparation, those steps necessary to put transaction data into usable form for processing and data entry, the activity of putting the data into the computer for processing.

The objectives of input design are

i) Controlling the amount of design.
ii) Avoiding delay.
iii) Controlling errors.
iv) Keeping the steps simple.

Controlling the Amount of Input

The major reasons for controlling the amount of input are, first, as the cost of labor is high, the cost of preparing the data and entering the data is also high .Second, the input phase of computing can be a slow process that can take many times longer than the time needed by computers to carry out their tasks. By reducing input requirements, the analyst can speed the entire process from data capture to processing to providing results to users.

Avoiding Delay

A processing delay results from data preparation or data entry operations is called a bottleneck. Avoiding bottleneck should always be one objective of the analyst in designing input.

Avoiding Errors

The third objective deals with errors .In one sense, the rate at which errors occur depends on the quantity of data ,since the smaller the amount of data to input, the fewer the opportunities for errors. The manner in which the data is entered also affects the occurrence of errors. Another aspect of avoiding error is the need to detect the error when they do occur. These are done using input validation techniques.

Avoiding Extra Steps

When the volume of transaction cannot be reduced, the analyst must be sure the process is as efficient as possible. The analyst must also avoid input designs that cause extra steps.

Keeping the process simple

Simplicity works and it is accepted by user's .In contrast it takes work to get users to accept complex or confusing input designs, and there is no guarantee of success in installing a complex system. In my module Inventory Management System all the above discussed points are given due importance while designing the input.

4.6.2 OUTPUT DESIGN:

Output design refers to the results and information that are generated by the system. Output is the main reason for developing the system and it is on the basis of the outputs generated that the usefulness of the application is evaluated. The significant points that have to be considered for the output design are as follows:

- Determine what information to present.
- Decide whether to display, print, or voice the information and select the medium of output.
- Arrange the presentation of information in an acceptable format.

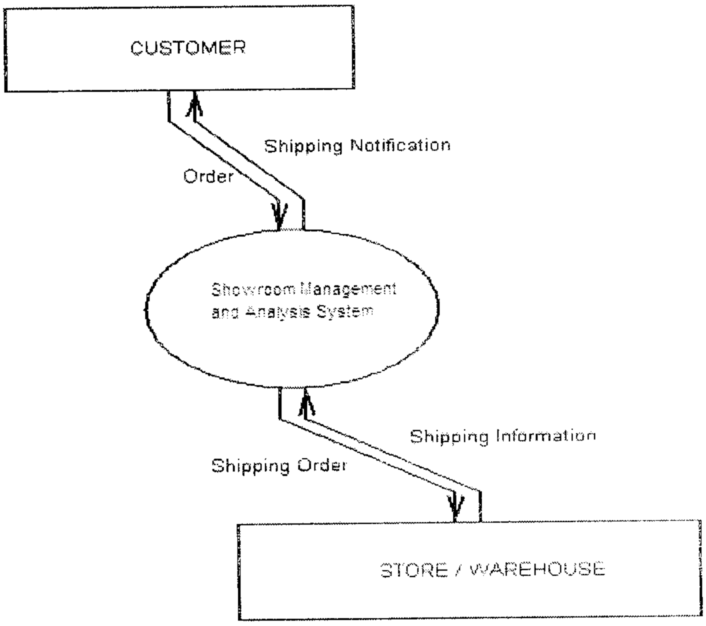
Decide how to distribute the output to intended recipients.

The arrangement of information on a display, or a printed document is termed as a layout. The output design is specified on layout forms, sheets that describe the location characteristics and the format of the column headings and pagination. Output design phase of system is concerned with the convergence of information to the end user in user-friendly manner. The output design should be efficient, intelligible so that the system relationship with the end user is improved considerably and thereby, enhancing the decision-making process. The contents of the output are then defined in a detailed manner during the physical design of outputs. Certain data are edited in a desirable manner.

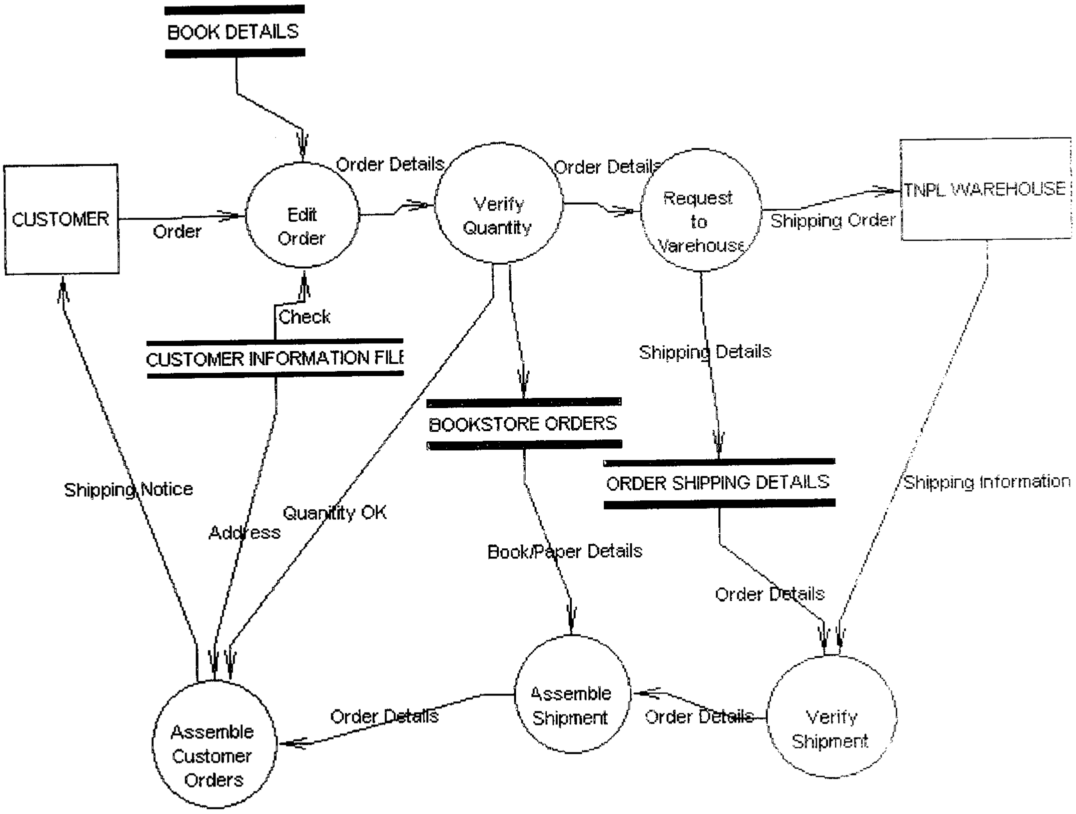
4.7 DATA FLOW DIAGRAM

This is one of the methods to show the flow of data within a module. The data flow represented by means of the arrow with the indication of the direction of flow with the help of the arrow pointer.

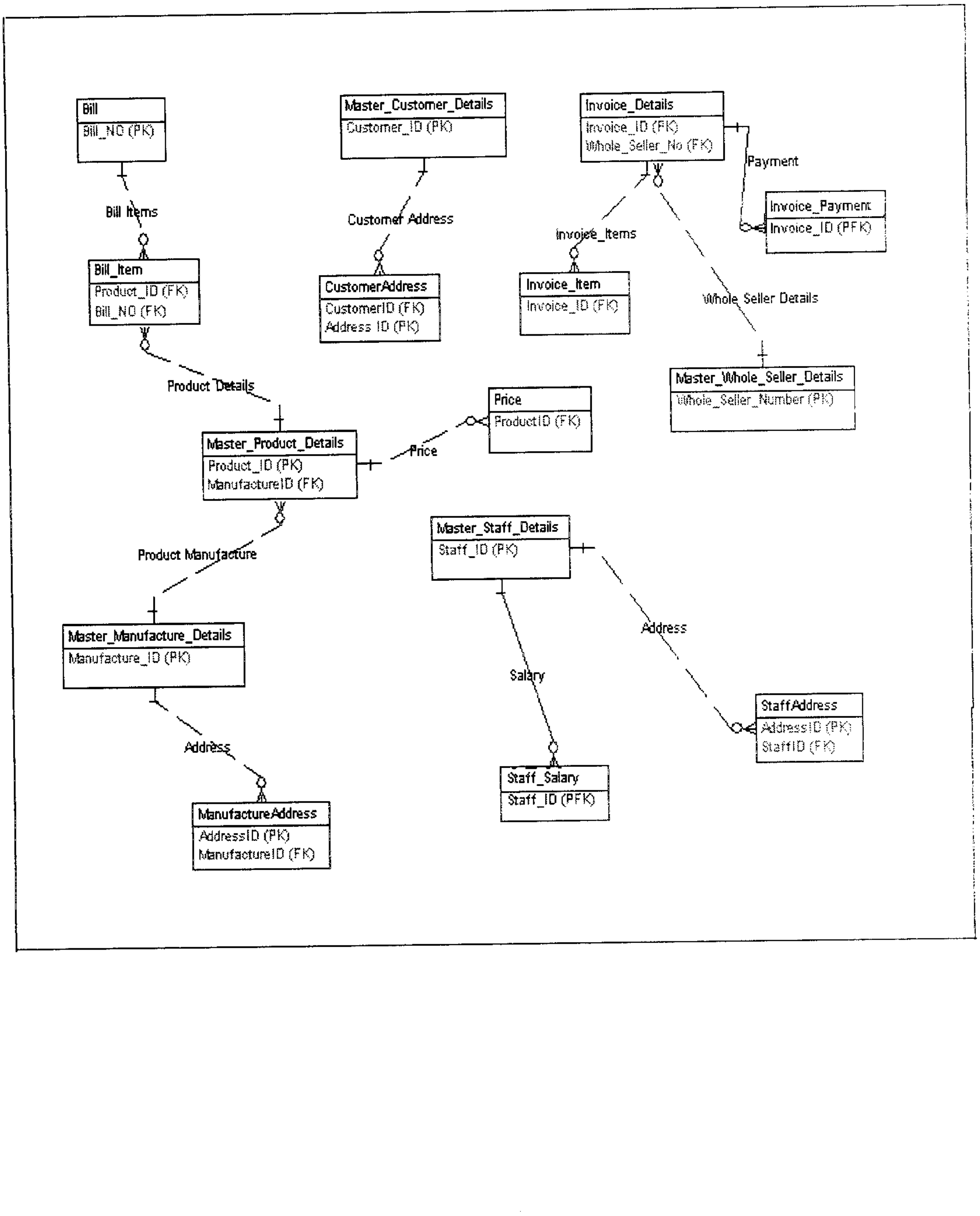
Diagram I



4.7.2 Data Flow Diagram



4.7.3 Entity Relation Ship Diagram



4.8 DATA DICTIONARY

4.1 TABLE NAME: BILL

FIELD NAME	KEY	DATA TYPE
Bill #	PRIMARY KEY	VARCHAR (10)
Customer #	FOREIGN KEY	VARCHAR (10)
Date		DATE

4.2 TABLE NAME: BILL ITEM

FIELD NAME	KEY	DATA TYPE
Bill #	FOREIGN KEY	VARCHAR (10)
Product #	FOREIGH KEY	VARCHAR (10)
Expiry		DATE
Quantity		INT
Amount		MONEY

4.3 TABLE NAME: INVOICE ITEM

FIELD NAME	KEY	DATA TYPE
Invoice #	FOREIGN KEY	VARCHAR(10)
Product #	FOREIGN KEY	VARCHAR(10)
Manufacture #	FOREIGN KEY	VARCHAR(10)
Batch		VARCHAR(10)
Expiry		DATE

4.4 TABLE NAME: CUSTOMER ADDRESS

FIELD NAME	KEY	DATA TYPE
Customer #	FOREIGN KEY	VARCHAR(10)
Address		VARCHAR(200)
City		VARCHAR(100)
Zip		NUMERIC(6)
Country		VARCHAR(50)
Office Phone		NUMERIC(20)
Home Phone		VARCHAR(20)
Email		VARCHAR(50)
Web		VARCHAR(50)

4.5 TABLE NAME: MANUFACTURE ADDRESS

FIELD NAME	KEY	DATA TYPE
Manufacture #	FOREIGN KEY	VARCHAR(10)
Address		VARCHAR(200)
City		VARCHAR(100)
Zip		NUMERIC(6)
Country		VARCHAR(50)
Phone		NUMERIC(20)
Home phone		NUMERIC(20)
Email		VARCHAR(50)
Web		VARCHAR(50)

4.6 TABLE NAME: STAFF ADDRESS

FIELD NAME	KEY	DATA TYPE
Staff #	FOREIGN KEY	VARCHAR(10)
Address		VARCHAR(200)
City		VARCHAR(50)
Zip		NUMERIC (6)
Country		VARCHAR(50)
Phone		NUMERIC (20)
Home Phone		NUMERIC (20)
Email		VARCHAR(50)
Web		VARCHAR(50)

4.7 TABLE NAME: INVOICE DETAILS

FIELD NAME	KEY	DATA TYPE
Invoice #	PRIMARY KEY	VARCHAR(10)
Whole Seller #	FOREIGN KEY	VARCHAR(10)
Duplicate Invoice		VARCHAR(10)
Date		DATE
Balance		MONEY
Notes		VARCHAR(500)
Amount		MONEY
Terms		VARCHAR(50)

4.8 TABLE NAME: INVOICE ITEM

FIELD NAME	KEY	DATA TYPE
Invoice #	FOREIGN KEY	VARCHAR(10)
Product #	FOREIGN KEY	VARCHAR(10)
Manufacture #	FOREIGN KEY	VARCHAR(10)
Batch		VARCHAR(10)
Expiry		DATE
Quantity		NUMERIC(5)
Sales Tax		NUMERIC(10)
Price		MONEY
Extended Amount		MONEY

4.9 TABLE NAME: INVOICE PAYMENT

FIELD NAME	KEY	DATA TYPE
Invoice #	FOREIGN KEY	VARCHAR(10)
Date		DATE
Mode		VARHCAR(10)
Amount		MONEY

4.10 TABLE NAME: MASTER CUSTOMER DETAILS

FIELD NAME	KEY	DATA TYPE
Customer #	PRIMARY KEY	VARCHAR(10)
Title		VARCHAR(3)
First Name		VARCHAR(100)
Second Name		VARCHAR(100)

4.11 TABLE NAME: MASTER MANUFACTURE DETAILS

FIELD NAME	KEY	DATA TYPE
Manufacture #	PRIMARY KEY	VARCHAR(10)
Business Name		VARCHAR(250)
Title		VARCHAR(3)
First Name		VARCHAR(100)
Second Name		VARCHAR(100)

4.12 TABLE NAME: STAFF DETAILS

FIELD NAME	KEY	DATA TYPE
Staff #	PRIMARY KEY	VARCHAR(10)
Title		VARCHAR(3)
First Name		VARCHAR(50)
Second Name		VARCHAR(50)
Position		VARCHAR(50)
Qualification		VARCHAR(30)

4.13 TABLE NAME: MASTER PRODUCT DETAILS

FIELD NAME	KEY	DATA TYPE
Product #	PRIMARY KEY	VARCHAR(10)
Product Name		VARCHAR(100)
Product Description		VARCHAR(200)
Manufacture #	FOREIGN KEY	VARCHAR(10)
Quantity Sold		NUMERIC(10)
Minimum Quantity		NUMERIC(4)
Price		MONEY
Notes		VARCHAR(500)

4.14 TABLE NAME: STAFF DETAILS

FIELD NAME	KEY	DATA TYPE
Staff #	PRIMARY KEY	VARCHAR(10)
Title		VARCHAR(3)
First Name		VARCHAR(50)
Second Name		VARCHAR(50)
Position		VARCHAR(50)
Qualification		VARCHAR(50)

4.15 TABLE NAME: WHOLE SALE DETAILS

FIELD NAME	KEY	DATA TYPE
Whole Seller Number #	PRIMARY KEY	VARCHAR(10)
Business Name		VARCHAR(200)
Title		VARCHAR(3)
First Name		VARCHAR(50)
Second Name		VARCHAR(50)

4.16 TABLE NAME: PRICE

FIELD NAME	KEY	DATA TYPE
Product #	FOREIGN KEY	VARCHAR(10)
Batch #		VARCHAR(10)
Quantity		NUMERIC(5)
Price		MONEY

4.17 TABLE NAME: STAFF

FIELD NAME	KEY	DATA TYPE
Staff #	FOREIGN KEY	VARCHAR(10)
Salary/Bonus		VARCHAR(1)
Date		DATE
Salary Amount		MONEY

CHAPTER 5

TESTING AND IMPLEMENTATION

5.1 TESTING

5.1.1 TESTING OBJECTIVES

1. Testing is the process of executing a program with the intent of finding an error.
2. A good test case is one that has finding an undiscovered error.
3. A successful test is one that uncovers an error.

5.1.2 UNIT TESTING

This test focuses on each module individually, (i.e.) each module is tested to ensure that the functionality of the module is proper as a unit. And that it meets the user requirements.

5.1.3 INTEGRATION TESTING

Integration testing addresses the issue associated with the dual problems of verification. After the software has been integrated a set of tests are conducted to check if the system is working well as a group.

The following are the types of Integration Testing:

1. Top-down testing.
2. Bottom-Up testing.

5.1.4 TOP-DOWN TESTING

This method is an incremental approach to the construction of the program structure. Modules are integrated by moving downward through the control hierarchy, beginning with program module. The module subordinates to the main program are incorporated into the structure in either a depth first or breadth first manner.

5.1.5 BOTTOM UP TESTING

This method begins with the construction and testing with the modules at the lowest level in the program structure. The bottom-up integration strategy may be implemented with the following steps:

The low-level modules are combined into clusters into clusters that perform a specific software sub-function.

5.1.6 VALIDATION TESTING

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

5.1.7 VALIDATION TEST CRITERIA

Software Testing and validation is achieved through series of black box tests that demonstration 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 requirements. Both, the plan and procedure are designed to ensure that all functional requirements are achieved, documentation is correct and other requirements are meet.

5.1.8 SYSTEM TESTING

System testing is series of different tests whose primary purpose is to fully exercise the computer based system. Although each test has a different purpose, all the work should verify that all system elements have been properly integrated and perform allocation function.

5.2 IMPLEMENTATION

Implementation includes all those activities that take place to convert from the old system to the new. The new system may be totally new, replacing an existing manual or automated system, or it may be a major modification to an existing system. Proper implementation is essential to provide a reliable system to meet the organizational requirements. Successful implementation may not guarantee improvement in the organizational using the new system, as well as, improper installation will prevent any improvement.

The implementation phase involves the following tasks:

- Careful Planning.
- Investigation of system and constraints.
- Design of methods to achieve the changeover.
- Training of staff in the changeover phase.
- Evaluation of changeover.

CHAPTER 6

CONCLUSION AND FURTHER SCOPE

6.1 CONCLUSION

The development of **SHOWROOM MANAGEMENT AND ANALYSIS SYSTEM**, the purpose of the system required is achieved. This software has received positive response from the user department due to following facilities available in the software such as invoice, purchase order processing, and purchase inward entry with tax, discount, batch number and expiry date etc. The Billing process work is reduced and several billing types are introduced. The SMAS helps to analyze the stock, sales and produce a report which helps the outlet management to make a better decision. The goals that have been achieved by the SMAS are

- ✓ User Friendly.
- ✓ Portable and flexible for further development.

6.2 FURTHER SCOPE

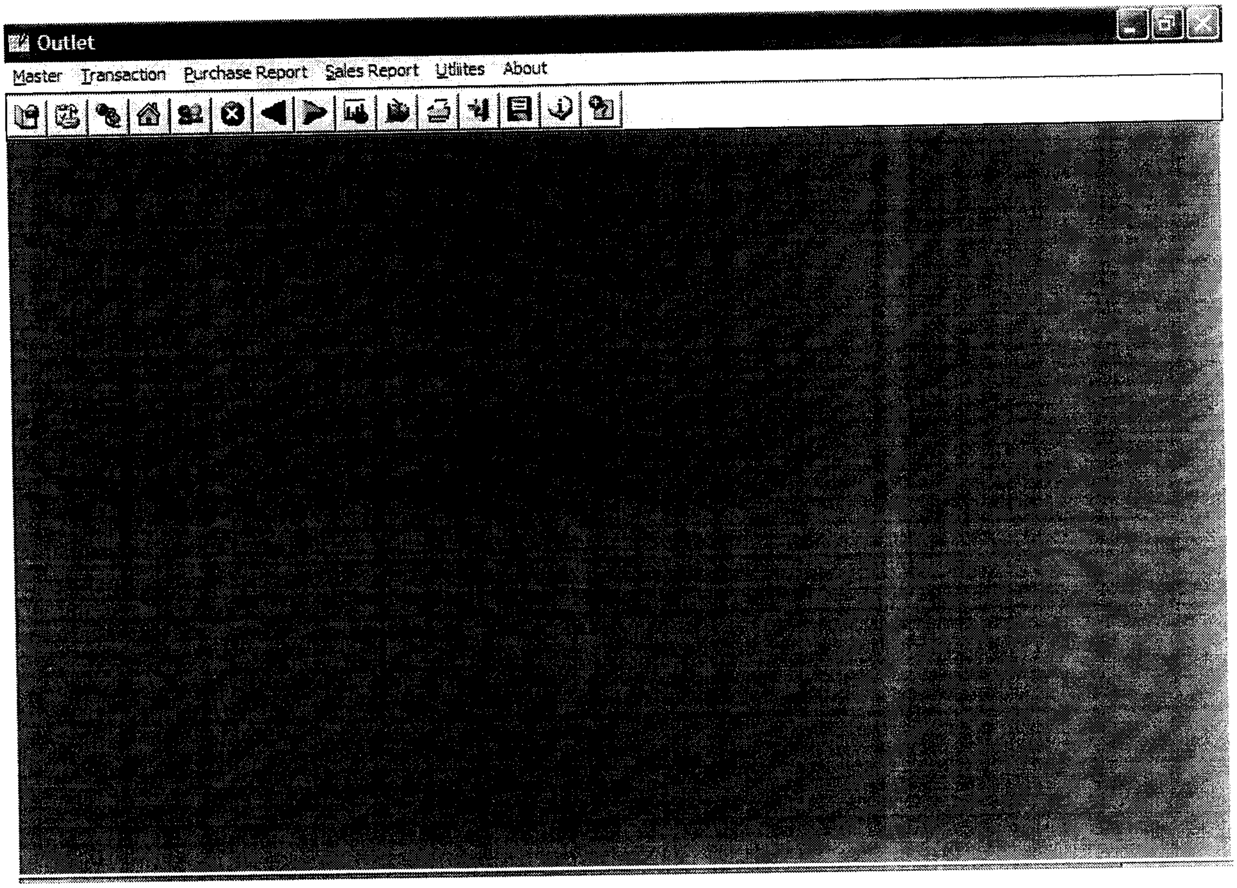
The below mentioned are few of the suggestions for further work in this development software:

- ➢ More customized reports can be programmed instead of the pre-defined reports.
- ➢ Integrating the all Outlet with TNPL using VPN.

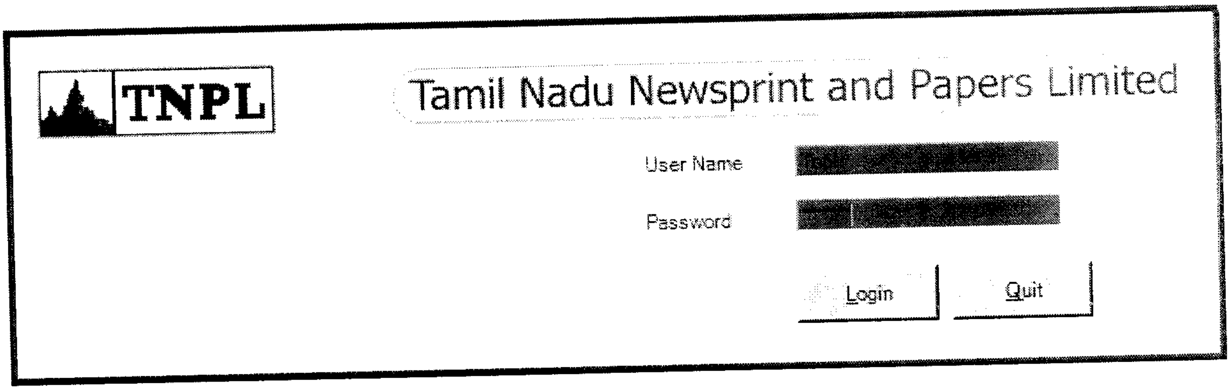
APPENDICES

SCREEN SHOTS

A.1 Main Screen:



A.2 LOGIN SCREEN

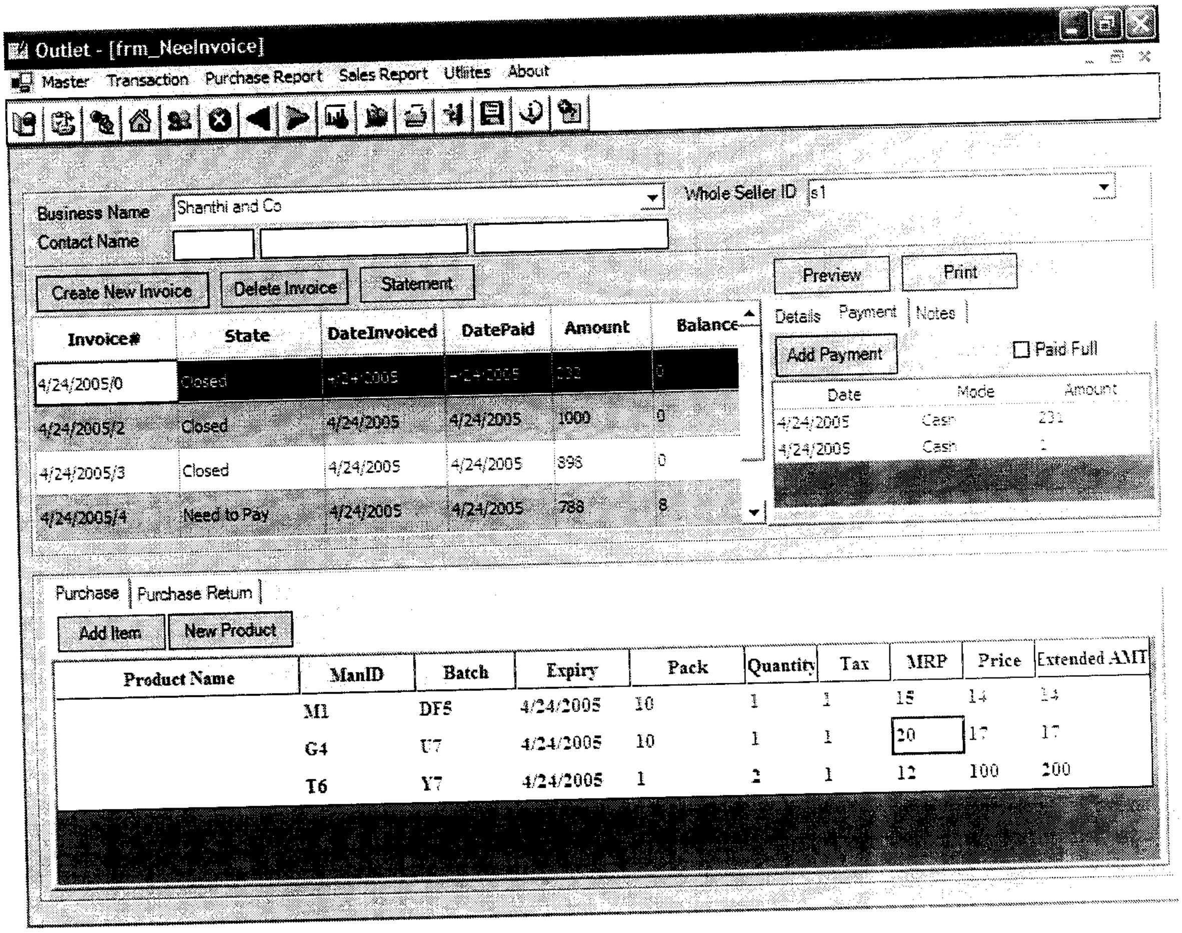


Tamil Nadu Newsprint and Papers Limited

User Name [redacted]

Password [redacted]

A.3 INVOICE FORM



Business Name: Shanthi and Co

Contact Name:

Whole Seller ID: s1

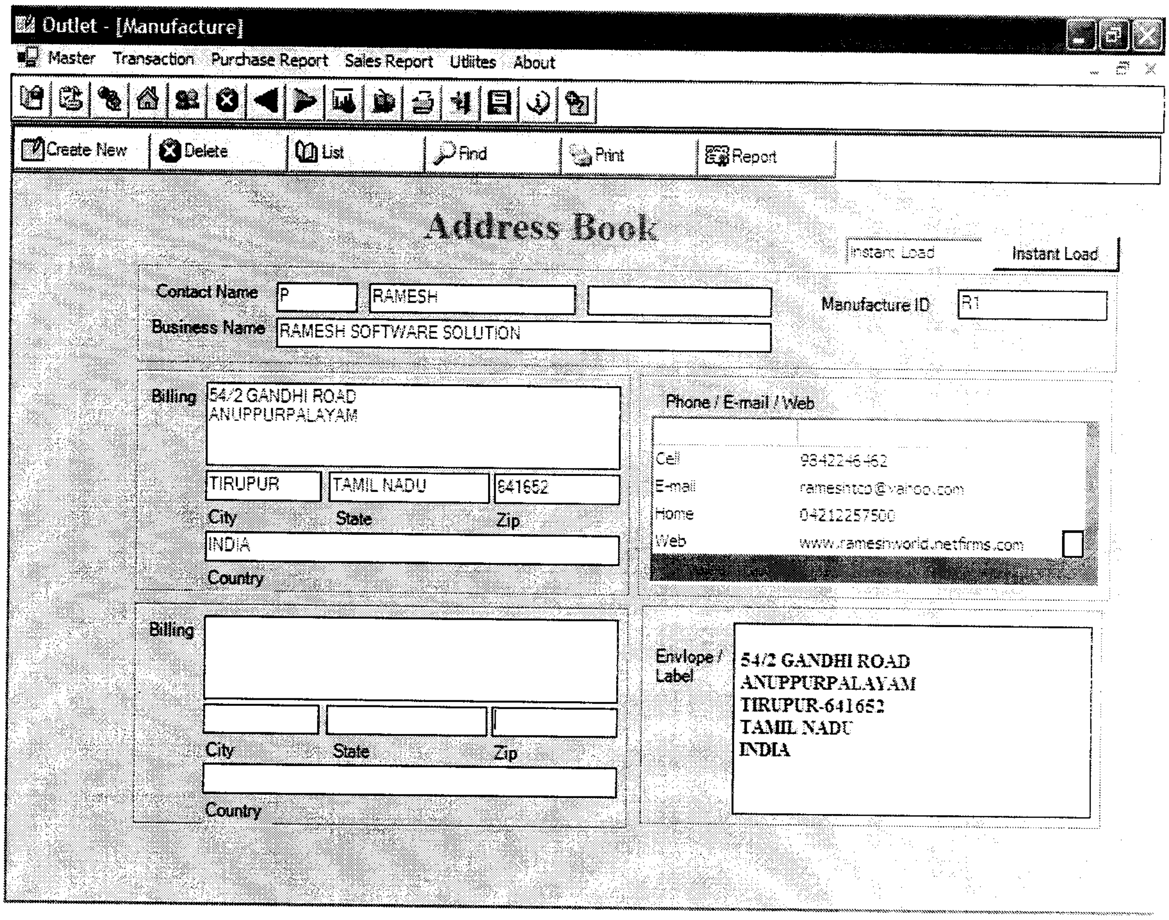
Invoice#	State	DateInvoiced	DatePaid	Amount	Balance
4/24/2005/0	Closed	4/24/2005	4/24/2005	232	0
4/24/2005/2	Closed	4/24/2005	4/24/2005	1000	0
4/24/2005/3	Closed	4/24/2005	4/24/2005	896	0
4/24/2005/4	Need to Pay	4/24/2005	4/24/2005	788	8

□ Paid Full

Date	Mode	Amount
4/24/2005	Cash	231
4/24/2005	Cash	1

Product Name	ManID	Batch	Expiry	Pack	Quantity	Tax	MRP	Price	Extended AMT
	M1	DF5	4/24/2005	10	1	1	15	14	14
	G4	U7	4/24/2005	10	1	1	20	17	17
	T6	Y7	4/24/2005	1	2	1	12	100	200

A.4 MANUFACTURING DETAILS FORM



Address Book

Contact Name	P RAMESH
Business Name	RAMESH SOFTWARE SOLUTION
Manufacture ID	R1

Billing
54/2 GANDHI ROAD
ANUPPURPALAYAM

City	State	Zip
TIRUPUR	TAMIL NADU	641652

Country: INDIA

Phone / E-mail / Web

Cell	9842246462
E-mail	rameshtco@yahoo.com
Home	04212257500
Web	www.rameshworld.netfirms.com

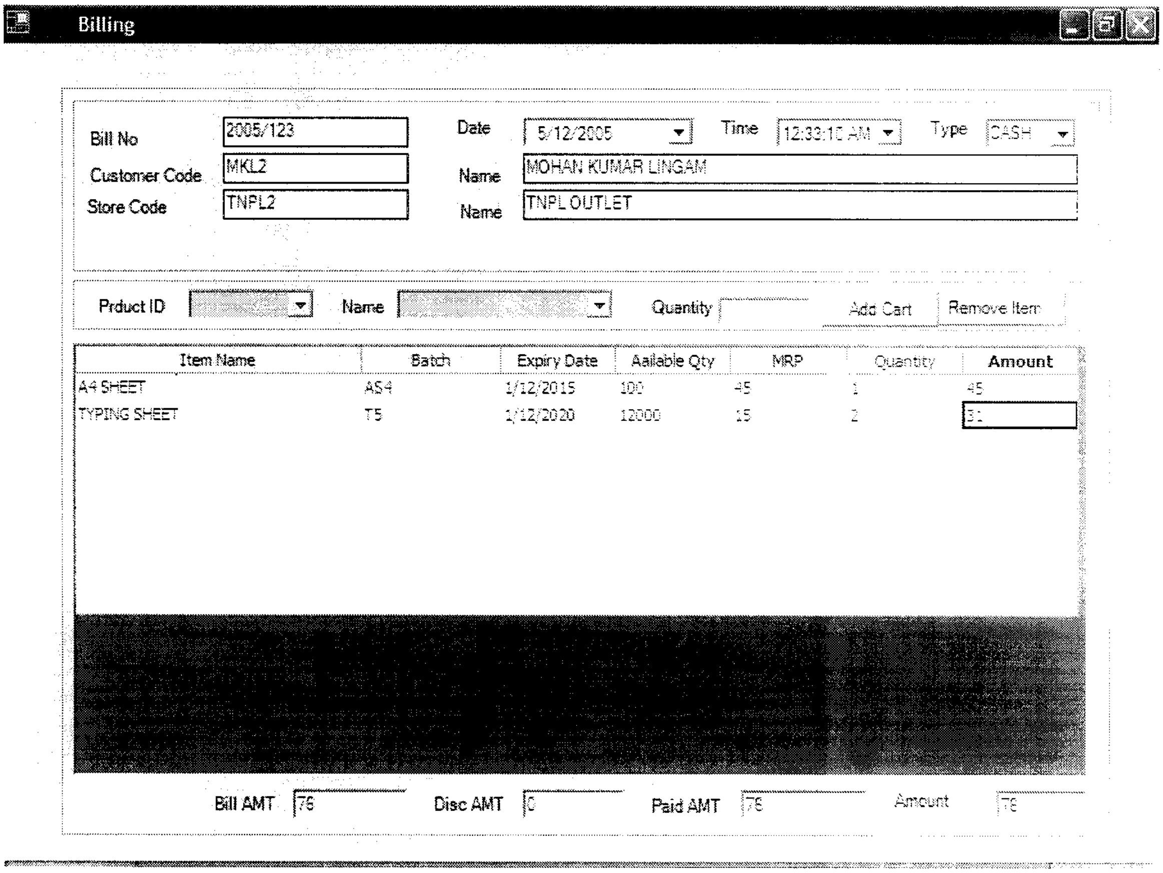
Billing

City State Zip

Country

Envlope / Label
54/2 GANDHI ROAD
ANUPPURPALAYAM
TIRUPUR-641652
TAMIL NADU
INDIA

A.5 BILLING FORM



Bill No 2005/123

Date 5/12/2005

Time 12:33:10 AM

Type CASH

Customer Code MKL2

Name MOHAN KUMAR LINGAM

Store Code TNPL2

Name TNPL OUTLET

Prduct ID

Name

Quantity

Item Name	Batch	Expiry Date	Aailable Qty	MRP	Quantity	Amount
A4 SHEET	AS4	1/12/2015	100	45	1	45
TYPING SHEET	T5	1/12/2020	12000	15	2	31

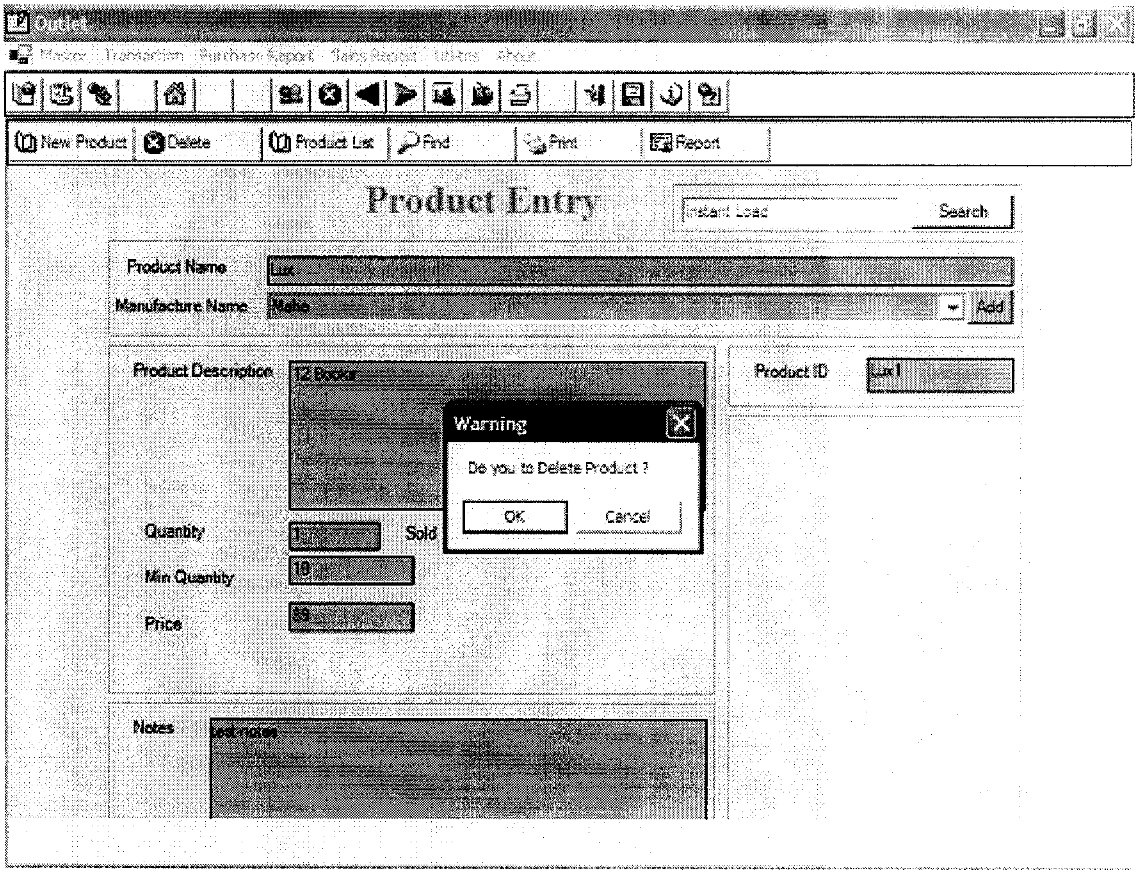
Bill AMT 76

Disc AMT 0

Paid AMT 76

Amount 76

PRODUCT ENTRY FROM



Product Name: Lux

Manufacture Name: Maho

Product Description: 12 Books

Product ID: Lux1

Quantity: 1

Sold

Min Quantity: 10

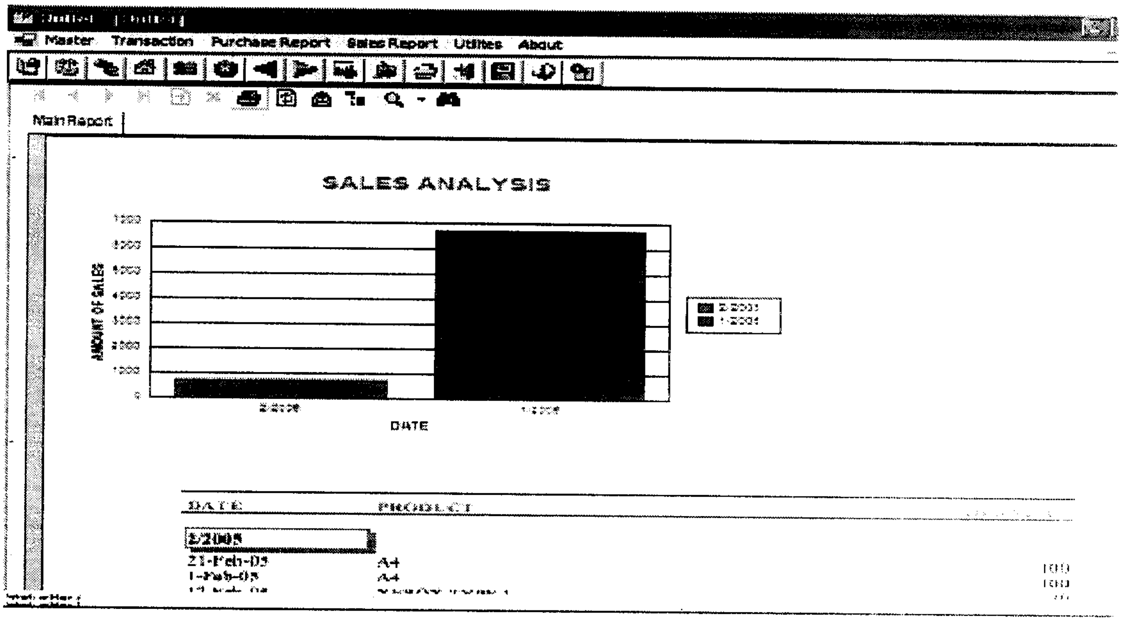
Price: 89

Notes: test notes

Warning

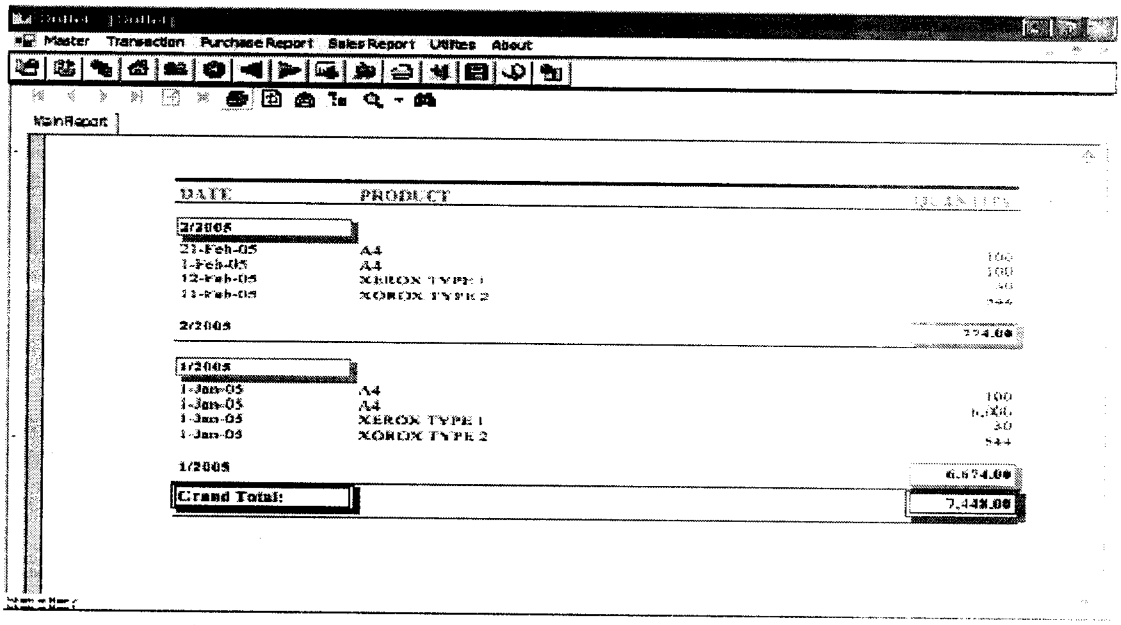
Do you to Delete Product ?

A.7 Sales Analysis Report:



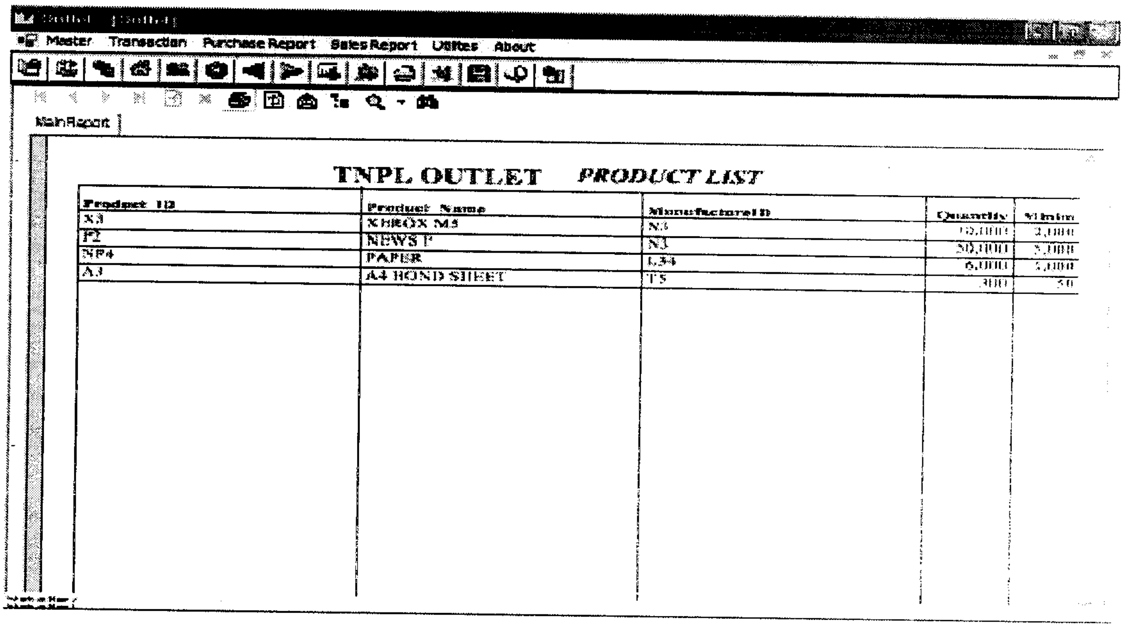
DATE	PRODUCT	[illegible]
2/2005		
21-Feb-05	A4	100
1-Feb-05	A4	100

A.8 Monthly Sales Report:



DATE	PRODUCT	QUANTITY
2/2005		
21-Feb-05	A4	100
1-Feb-05	A4	100
12-Feb-05	XEROX TYPE 1	30
11-Feb-05	XOROX TYPE 2	544
2/2005		774.00
1/2005		
1-Jan-05	A4	100
1-Jan-05	A4	6,000
1-Jan-05	XEROX TYPE 1	30
1-Jan-05	XOROX TYPE 2	544
1/2005		6,674.00
Grand Total:		7,448.00

A.9 Product List Report:



TNPL OUTLET ***PRODUCT LIST***

Product ID	Product Name	ManufacturerID	Quantity	Minim
X3	XEROX M3	N3	10,000	2,000
P2	NEWS P	N3	20,000	5,000
NP4	PAPER	L34	6,000	5,000
A3	A4 BOND SHEET	T5	300	50

REFERENCES

Books

[1] Gray Cornell "Programming VB.NET ,A guide for Experienced Programmers" , Apress Publication

[2] Damein Watkins "Programming in the .NET Environment", 2003 BPB Publications

[3] Ron Sonkup and Kalen Dalaney "Inside Microsoft SQL Server 2000", 2000 Microsoft Press

URL

http://www.microsoft.com

http://www.vbdotnet.com