COMMITMENT MONITORING SYSTEM

PROJECT WORK DONE AT

Ashok Leyland Ltd.,
Chennai.

P-194

PROJECT REPORT

Submitted in partial fulfillment of the
Requirements for the award of the degree of
Master of Computer Applications
of Bharathiar University

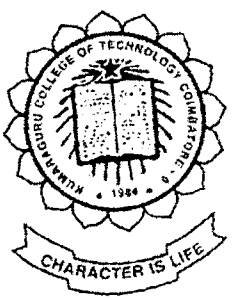
Submitted By

P. Suresh Kumar
Reg. No. : 9938M0651

Guided By

EXTERNAL GUIDE
Ms. MADAVI LATHA

INTERNAL GUIDE
Ms. PARAMESHWARI MCA



Department of Computer Science and Engineering
Kumaraguru College of Technology
Coimbatore – 641 006
May 2002

Department of Computer Science and Engineering

Kumaraguru College of Technology

(Affiliated to Bharathiar University)

Coimbatore – 641 0060000

CERTIFICATE

This is to certify that the project work entitled

"Commitment Monitoring System"

Done By

P.SURESH KUMAR

Reg. No. : 9938M0651

Submitted in partial fulfillment of the requirements for the award of the degree of

Master of Computer Applications of Bharathiar University.

S. Jha 30/4/02

Professor and Head

Internal Guide

Submitted for the University Examination held on 10-05-2002

10/5/02

Internal Examiner

External Examiner

Ashok Leyland

BUILDING No.2, 7th FLOOR,
KHIVRAJ COMPLEX,
477-482, ANNA SALAI,
NANDANAM,
CHENNAI - 600 035.
PHONE : (044) 4331120
FAX : (044) 4338344
WEB : www.ashokleyland.com

April 12, 2002

PROJECT COMPLETION CERTIFICATE

Name	:	P Suresh Kumar
Register Number	:	9938M0651
Course	:	Master of Computer Applications
Institution	:	Kumaraguru College of Technology Affiliated to Bharathiyar University, Coimbatore – 641 006.
Date of Joining	:	10th December 2001
Date of Completion	:	11th April 2002
Department/ Division	:	Systems Department, Corporate Office
Project Work	:	Commitment Monitoring System
Remarks	:	Very Good

For Ashok Leyland Limited

SHEKHAR ARORA
GM – Management Development

DECLARATION

I hereby declare that the project entitled "**Commitment Monitoring System**" submitted to Bharathiar University as the project work of Master Of Computer Application Degree, is a record of original work done by me under the supervision and guidance of Ms.R.Madavi Latha, Executive, Systems, Ashok Leyland Ltd. Chennai and Ms.Parameswari, Lecturer, Department of Computer Science and Engineering, Kumaraguru College Of Technology, Coimbatore and this project work has not found the basis for the award of any Degree/ Diploma/ Associate ship/ Fellowship or similar title to any candidate of any University.

(P. Suresh Kumar)
Reg. No.9938M0651

Place: Coimbatore

Date: 29-04-2002

Countersigned By

(Internal Guide)

ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion. I wish to express my sincere thanks to all those who were involved in the completion of this project.

I am extremely grateful to **Dr.K.K.Padmanabhan,**Ph.D., Principal, Kumaraguru College of Technology for having given me an opportunity to serve the purpose of my education.

I am indebted to Prof. **Dr.S.Thangasamy**,Ph.D., Head of the Department of Computer Science and Engineering, for his valuable guidance and useful suggestions during the course of the project.

I am deeply indebted to my project guide, **Ms.Parameswari**,M.C.A Lecture of Department of Computer Science And Engineering, Kumaraguru College of Technology for her helpful guidance and valuable support given to me throughout the project.

I extend my most sincere thanks to **Mr.V.Varadharajan**, Senior Manager, Systems Ashok Leyland Ltd., for his suggestions and advice from the joining to throughout the project work.

In particular, I owe no end of thanks to my project guide **Ms.R.Mathavi Latha,** Executive, Systems, Ashok Leyland Ltd., for her enthusiastic guidance and encouragement throughout the project work.

I also express my sincere gratitude to Mr.K.Hariharan, Mr.S.Subramaniam, and Mr.M..Harikrishnan for their valuable guidance, support rendered and the courtesy extended to me during the tenure of my project in their esteemed organization.

Finally I records my thanks to my friends Ms.Niranjana, Mr.Srinivasa Manikandan, Mrs.Deepa , Mr.Umamageswaran , Ms.Vadivu, Mr. John leo britto and

SYNOPSIS

The main objectives of the **"Commitment Monitoring System"** is to automate the project & Taxation Department tasks. The project & Taxation Department of Ashok Leyland Ltd. Is responsible for preparing the purchase order and financial capital expenditure (Fincap). To know the status of fincap it is necessary to have a better control over project & Taxation Department operation.

This software is to minimize the workload and to get accurate and timely results. This software is very easy to operate. This software can be executed in IBM compatible machines under windows environment. This software was developed using the ORACLE8i with DEVELOPER 2000 / FORMS6i WINDOWS 98.

The following chapter gives a brief introduction about the project work. The remaining chapters are discussing about the concepts of this system, system requirements, language description and other details. The system is user friendly. The system was tested with real data and it was successful.

TABLE OF CONTENTS

TABLE OF CONTENTS

INTRODUCTION

1. INTRODUCTION

1.1 PROJECT OVERIVEW

In recent days the computer technology is developing more rapidly in all walks of life. This is due to the fact that, the necessity has arisen to store the information available in the organization in proper, accurate and in speedy manner.

The project entitled **"Commitment Monitoring System"** is done for the company Corporate Department of ASHOK LEYLAND LIMITED. The project is being implemented in Developer 2000, which is the front end, oracle8i acts as the backend.

Here in this project the financial capital expenditure (Fincap) is automated. The company finds it difficult to maintain the purchase order details, Fincap details and also the approval of the fincap by various departments. So the higher authorities decided to automate these operations.

Here in this project the details about the program they put for the each year is maintained, the various orders put for each program number is maintained and also the fincap for each order is also automated. The approvals of the fincap by various authorities are also automated. Various reports are generated which gives the details about the status of the fincap, Various fincap which are yet to be approved by so and so person etc.,

1.2 PROBLEM DEFINITION

Problem is a question requiring a solution or a difficult situation. Definition is exact description or explanation or meaning of a difficult term. Problem definition together as a phrase can be started as explaining a question or describing the difficult situation.

In viewpoint of our project until the job is manual we have problem and we need not define to or manual processing into computer automated process. When converting from one system to other it is better to ask questions against each existing action and describe it in an understandable manner.

- Objective: To make reports correlating the Fincap, purchase order and program details and the status of the Fincap.
- Tables:Program_Master,Po_Master,Po_Detail,Fincap_Master,Fincap_detail,Modification_Master,Modification_detail,Login,Approval
- Programs:ProgramCreation,Order Creation,Fincap Creation, Approval Creation,Approvation Information Creation, Program wise report,Fincap Status Reports etc.,

PROGRAM CREATION

This contains the details of the program, which is put every year ,like Program number, Program Name, Program Date, Total approval, Cleared For Commitment, Master Plan Forecast, apex From start, Capex Year To Date,Fincap Number Change Flag,Fincap Number.

ORDER CREATION

This contains the details of the order, like program number, purchase order, purchase order date, type of order, supplier name, location, type of job, type of sub job, order value, Different extra charges for local and imported orders etc.

FINCAP CREATION

This contains the details of the Fincap, like program number, fincap number, order number, order value, all details of the orders and the program and also details about the R & D check list and the previous release for the program and also for the particular job.

APPROVAL CREATION

This contains the details of the entity order and fincap in a single screen and it will be viewed and approved by different authorities.

REPORTS CREATION

Various details regarding the transaction are provided and the corresponding reports are generated.

1.3 ORGANIZATION PROFILE

Ashok Leyland is one of the largest manufactures of commercial vehicles and diesel engines in India. LRLIH Limited - which holds the majority shareholding of Ashok Leyland, previously owned by the Rover group, UK (erstwhile Leyland Group) - was acquired in 1987 by a joint venture of the **Hinduja Group** and **IVECO**, a fully-owned subsidiary of FIAT, heralding a new chapter in the history of Ashok Leyland.

The company was promoted as Ashok motors limited in 1948 to assemble the Well-known **Austin cars**. During 1950, the company commenced assembly and distribution of Leyland commercial vehicles. An agreement was concluded with Leyland Motors Limited. UK, to manufacture the vehicles in India and to participate in the equity capital of the company, which was then renamed Ashok Leyland limited.

Ashok Leyland vehicles have established a reputation for their reliability, superior performance and durability – 29 basic types of medium and heavy-duty vehicles are offered in different models to meet diverse applications in moving men and material.

Ashok Leyland has six manufacturing plants – located at Ennore, Hosur (Tamil Nadu), Bhandara (Maharashtra), Alwar (Rajasthan), and Hyderabad (Andhra pradesh) – with a total employment of more than 15,000.

HINDUJA GROUP

A major transnational conglomerate, the Hinduja Group operates in five continents in the following areas.

- International Trade and Marketing
- International Finance, Investment Banking and Asset management
- Manufacturing industries
- Industrial project Development
- Charitable Activities – Hinduja foundation

IVECO

IVECO, a fully – owned subsidiary of FIAT Italy, is a major force in the international commercial vehicle markets. IVECO was formed in 1975 by the merger of five European companies: FIAT VEICOLI INDUSTRIALI, OM and LANCIA VEICOLI SPECIALI of Italy, UNIC of France and MAGIRUS of Germany.

Integrating and rationalizing the production facilities, some were identified to specialize in different product segments – others to produce components and supply throughout the company.

One in every five new trucks and buses sold in Europe is mad3 by IVECO. It is one of the world's top manufactures of diesel engines. IVECO has been, for long, Europe's leading exporter of commercial vehicles.

It also has a considerable manufacturing presence outside Europe:

- In **China**, with the **Nanjing** motor corporation.
- In **India**, with **Ashok Leyland**.
- In **Australia**, with **international Trucks**.
- In **Korea**, with **Halla**.
- In **Vietnam**, with **Mekong corporation**.

Ashok Leyland

The company produces chassis for different types of medium and Heavy Duty Vehicles in different models. The vehicles are exported to 27 countries round the world from Philippines to the Caribbean.

In this goods range, Ashok Leyland Ltd., offers vehicles ranging from 7.5 T GVW to 125 T GVW.

This includes,

a) Taurus – the only 3-axled on-the road vehicle manufactured in the country,
b) Rhino 6 x 4.
c) Turbo Tractor – India's largest trailer.
d) Titan – the Double Decker and other specially designed vehicles for earthmoving, oil fields, defense and aircraft refueling operations.

THE DEPARTMENTS

Ashok Leyland Limited is managed by a Board of Directors. Sales, Service and Parts network covers almost all the big cities of the country. The various manufacturing responsibilities within the overall corporate parameters are like Production, Materials, Finance, Personnel, Quality, and Industrial Engineering etc.

MANUFACTURING

This division performs crucial task of producing quality products interpreted in a hierarchical fashion as follows

- Vehicle : Different models of chassis for a variety Applications.
- Major Units : Engines, Front & Rear axles, GearBox and Frames.
- Sub-Assemblies assembly : Hand brakes, Clutches, propeller Shaft, oil pump etc.

MATERIALS

This Department is responsible for planning requirements of materials and at optimum costs, so as to ensure the uninterrupted manufacturing flow. The Materials Department plays a major role in managing corporate profitability since,

- Material costs determine the price structure of end product and margins thereof.
- Inventory costs determine the working capital requirements and costs thereof.
- Availability of materials at the right time is essential for optimization of production capacity.

SALES AND MARKETING

The various ranges of medium duty and heavy-duty vehicles, quality spare parts, lay parts are sold by the sales and marketing divisions to the domestic and overseas customers either directly or through dealers.

PRODUCT DEVELOPMENT

The customer needs are the major factors guiding the product development. A major portion of the product development at Ashok Leyland has been directed towards evolving concepts, which will bring in greater advantages to the operator. The emphasis is on fuel efficiency, cost reduction, and using the emerging latest technology.

The various other departments are the Industrial engineering, the Project planning and co-ordination, Corporate business planning, Finance, Management services each assuming the concerned responsibilities.

THE PRODUCTS

The products of Ashok Leyland Ltd. Can be grouped into four major classes such as Passenger Vehicles, Goods Vehicles, Marine engines and

System study and Analysis

2. SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM

The existing system is manual. IT is highly tough for the administration people top maintain.

- Data security is less
- It involves lot of manual work
- The data are stored in manuals registers and requires frequent updating this is very impossible
- Duplication cannot be avoided
- It is time consuming process

Being a manual system it is very tedious work of going through ledgers to find out the even simple task of balance available on a particular job etc. It also involves tedious work of posting of values in different heads in which there might be a wrong posting.

Being such a tedious work it is not an easy job to do it manually. It was a tedious process and it required large amount of time and more over it was a very complex process. Moreover when a fincap is to be processed several ledgers were to be referred which is a very complex and unsecured process. So the existing system is not much suited for the application.

Also it was a very complicated work to send the entire fincap manually for authorization. Therefore these were automated to reduce the effort, work, time and complexity. To overcome the above difficulties the Commitment Monitoring System was automated. The proposed system would overcome the difficulties like

- Time consuming
- May cause many errors
- More work
- Inconsistent

2.2 PROPOSED SYSTEM

Since the processing of such a large number of applications in the limited time available is not possible, and the number of errors committed in a manual system could be large, the system needs automation. Automating the system will result in

- Reduction of errors
- Speedy calculation
- Less processing time
- Paperless office
- Reducing the processing time
- Avoidance of duplication in allocating control numbers
- For better Management control

To overcome all the difficulties in the existing system, the proposed system has been developed in such a way that it would eradicate all the difficulties mentioned in the existing system. In order to carry out the work quickly, efficiently and within a specific time this system was developed. This system is designed to allow the unit to keep in track of the program details. Fincap details and Fincap approved details, note to MD details, Modification of purchase order details, deviation request, findisposal, findisposal approved details.

2.3 REQUIREMENTS OF NEW SYSTEM

Functional Requirements

Introduction.

The main objective of developing this project of Commitment Monitoring System is to simplify the process of maintaining the database of program details, order details, fincap details etc.

List of Inputs

This system divided in to four modules

1. Program Entry.
2. Order Entry.
3. Fincap Entry.
4. Approval.

Processing Requirements

This Commitment Monitoring System has four Processing systems.

Program Processing

Program processing deals with the creation of the program number every year, for which an value limit will be decided and that amount of value will be used to put orders.

Order Processing

Order value will be subtracted from the value limit. Once the order value exceeds the value limit the we cannot put an order further for that particular program number.

Fincap Processing

In the fincap processing a fincap will be processed for each order number and many details about the fincap is also processed.

Approval Processing

Many divisions will view the fincap. They view all the order details, program, fincap details and authorize it and send it to the next level for authorization.

Reports

Pending fincaps details.

Approved fincaps.

Purchase order details.

Findisposal details.

Performance Requirements

Security

The security requirements are listed below:

✓ Each user is given a user id that identifies the user to the dbms software. So every user on the database is permitted based on whether the user is a valid user or not.

✓ When the user view the records the update and delete operations are unavailable.

✓ Privileges. These are set of actions that a user can carry out against database objects. The privileges provided to user are select, insert, delete, and update privileges. Privileges once granted can also be revoked.

Availability

All users in the Ashok Leyland corporate department can use this package. The database server always available. So the user can use the system at any time.

Capacity

This application run only the environment of oracle and developer 200.We can append the required module to this application.

Response Time

The response time depends upon the size of the database and its structure, network traffic, the speed of the processor.

Oracle is an easy to use interface and helps the user to be more productive. By providing appropriate tools for different aspects of GUI environment. The screens are option driven which help us to navigate between them easier. The input can be given on data entry screens with the cursors getting positioned automatically to the next data after the pressing of every tab key.

Data should be entered for all the fields. For the modification of the fields the query button in the toolbar should be pressed to retrieve the old data all the data are validated and if there is an invalid data a message box is flashed in the screen. This enables users to interact with the application.

As far this project is concerned user interface is designed in Developer 2000. Good software should have a best user interface facility in order to get the maximum utilization of the software in effective and in efficient manner and also to get the maximum satisfaction of the user.

The user had the different kinds of interface requirements in the "**Commitment Monitoring System**". The first important requirement is the screens should be as much as elegant and the screen should not be a confusing one. The error messages and the messages regarding the updating of data have to be clear.

Data should be entered for all the fields. For the modification of the fields alter key should be pressed to retrieve the old data and if new data, Create key should be pressed and then a new data is entered. All the data are validated and if there is an invalid data a message box is flashed in the screen.This enables users to interact with the application.

Programming Environment

3. PROGRAMMING ENVIRONMENT

3.1 HARDWARE ENVIRONMENT

The hardware environment in which the software has been developed can be quoted as below:

SERVER

Processor	: Pentium – III
Clock Speed	: 800MHz
Hard disk capacity	: 20GB
RAM capacity	: 128MB
Floppy disk	: 1.44 MB
Keyboard	: Windows Keyboard
Monitor	: 15 inches Compaq

CLIENT

Processor	: Pentium – III
Clock Speed	: 500MHz
Hard disk capacity	: 10.2GB
RAM capacity	: 64MB
Floppy disk	: 1.44 MB
Keyboard	: Windows Keyboard
Monitor	: 15 inches Compaq

3.2 DESCRIPTION OF SOFTWARE AND TOOLS USED

Software Environment

O/S on Server	: Windows 2000
O/S on Client	: Windows 98
RDBMS	: Oracle 8*i*
TOOLS	: Developer 2000 / Forms 6*i*, Reports 6*i*

ABOUT THE SOFTWARE

Client/Server computing is now all the rage in the corporate data processing world. A Client/Server application is an application that you divide into at least two parts, one of which requests things from the other. The Requester is the Client and the other is the Server.

Clients can request anything from a Server, behavior or data. The only important thing is about the relationship. The essence of Client/Server computing is the distribution of work over two or more computers. A key facility of Client/Server computing is the ability to partition the application between the client and server.

There are two types of clients, namely, thin client and fat client. A *fat client* puts virtually all the application processing on the thin client. A *thin client* moves much of it to the server. *Developer/2000 can* be of either kind of client, depending on where the procedure code appears in the application triggers or in database triggers and procedures.

Developer/2000 provides us with all the tools we need to develop robust, scalable, and fast applications in a client/server or World Wide Web environment, especially when we use *Oracle 8i*

DATA BASE APPLICATIONS

A *Database manager* is a software system that lets us query, manipulate and control data through the database language *SQL*. When we use a database manager with our application, we create a database application rather than just an application.

A database application is a program that uses data from a database management system such as *Oracle 8i*. An *application generator* is a tool that builds applications mainly through declarative specification of the application rather than through procedural programming.

BACK-END TOOL: ORACLE

The oracle server is a state of the art information management environment. It is repository for very large amounts of data, and gives user rapid access to that data. The Oracle allowed for the sharing the data between applications; the information is stored in one place and used by many systems.

The Oracle server runs on dozen of different computers, supporting the following configurations:

- Host based
- Client/Server
- Distributed Processing
- Web-Enabled Computing

Components of Oracle Server

The base product of Oracle provides all the functionality to support the requirements of most Oracles' customer. The procedural component of the Oracle server delivers.

- PL /SQL
- Stored Procedures
- Database Triggers
- Packages and functions.

Distributed processing component

The distributed component allows users to work with data remote database as if resided locally.

ORACLE TOOLS

SQL * PLUS

- SQL *plus is an incentive program use to access an oracle database.
- It is also a database administrator's best friend and indispensable tool.
- Server Manager
- Worksheet Utility in enterprise manager
- PL/SQL is a procedural language to Oracle in extension to ANSI standard.
-

ORACLE FORMS

Oracle forms are feature rich application building tool that produces production quality screens utilizing data stored in a database. It can embed graphics, sound, video, word processing documents & spreadsheets.

ORACLE REPORTS

It produces production quality outputs using data sources such as the oracle database. We can embed graphics, sound, video, & wide assortment of visual aids in screen and hard copy output.

ORACLE LOADER

Oracle loader reads files & places the data in the Oracle database based on the instructions it receives on a control file.

OCL

Oracle Call Interface is a call interface to an Oracle database, which allows users to embed Oracle calls directly in high-level languages such as C, FORTRAN or COBOL.

FRONT END TOOL: DEVELOPER 2000- Forms6i

Developer 2000 is an ideal programming language for developing sophisticated professional applications for Microsoft Windows. It makes use of graphical user interface for creating robust and powerful applications.

This feature makes it easier to comprehend things in a quicker and easier way. It has other features like easier comprehension, user-friendliness, and faster application development.

Developer 2000 for windows requires at least Microsoft Windows 95/98/NT, 486 processor and a minimum of 16 MB of RAM. It has different versions like FORMS 4.5, 6.5, 6i. It has different version for generating reports – REPORTS 6i. It is made up of number of components like

- Menu bar
- Tool bar
- Properties Window
- Form Layout window
- Toolbox
- Form designer

FORM APPLICATIONS

The first version of the *Forms* component of *Deverloper/2000* was the *Interactive Application Generator, IAG.* Oracle then began to use the term 'forms' as part of the product name: *SQL*Forms.*

A form application is thus an application that presents data in an online format consisting of a series of fields laid out in one or more windows. This provides a food way of viewing the information that the application contains, entering and changing that information. It also provides the development menu and PL/*SQL* library modules.

The form module is the main component of the interactive applications. It is also the most complex module in terms of internal structure, containing many separate kinds of objects.

A form application can have one or more forms, depending on the tasks it has to accomplish. Form designing in the right format or the form becomes difficult to use and error prone. Breaking it into sections is a common way to reduce the complexity of a very large and complicated form. The tools in a form let us manage the records (query, insert, update, and delete, scrolls and so on).

TRIGGERS

A trigger is a block of PL/SQL we attach to another object: a form, a data block, or a data block item. We can also have triggers on property classes. The trigger *'fires'*, or executes, when certain events occur; the event triggers the code.

MASTER – DETAIL RELATIONSHIP

There is a particular kind of form called a *master-detail* form that divides the form into a smaller record and several detail records. This organization is so common that most sophisticated generators handle it as a specific option for building the form.

Applications of the *Developer /2000* forms let us build application without diving into the depths of a programming language and the accompanying *graphical user interface* (GUI) frameworks. These applications use already debugged code that does what it should do.

The current combination of power, safety and flexibility provides the right mix for productive application development. By abstracting the most important parts of application building into powerful commands and data structures application generators can quickly build applications that would take days or weeks to build in a programming language.

REPORTS APPLICATIONS

A report is a page-oriented display of data. Where as form provides an interactive tool for managing data, the purpose of a report is to format a large amount of data in a readable fashion.

Developer /2000 provide us with many different kinds of reports:

- Tabular: A simple table of data.
- Mailing label: A series of regularly repeating records formatted on each page in a certain area.
- Form letter: Boilerplate text surrounding data from a record that fills in

- Master – Master: Two groups of unrelated records displayed together.
- Master – Detail: A master record with two or more related detail records displayed together.
- Matrix – Cross tab: A cross – tabulation of two columns showing some aggregate or other value for the combination of each value from each column.
- Data File: A comma – separated or other variety of delimited data file; we use this to transfer data into other tools.

GRAPHICS APPLICATIONS

Graphics are pictorial representation of data. *Developer /2000* give us a sophisticated set of tools for both creating standalone graphics applications and for including graphics in the forms and reports. A display module may be a one or more chart we derive from database data, or it may contain any combination of graphic elements with or without reference to the database. The layout contains the graphic elements of the display in a hierarchy that represents their relationships. It also relates to those elements to the column query.

A *chart template* is a customized set of options that lets us build several charts that all have identical formatting possibly in the same display. A *query*, as in a report module, is a *SQL SELECT* statement that defines a set of data to display formatted as a chart.

INFERENCES ABOUT THE SOFTWARE

Developer /2000 are more than just the objects that comprise it. It is a system that glues together all of these objects into a working whole.

Not only do all these objects work together, but we can also work with objects from other applications and with data from database managers other than *Oracle 8i*.

The primary glue for Developer /2000 is PL/SQL, the programming language we use in triggers, program units, and other programs objects in Developer /2000. By accessing items and parameters PL/SQL is what makes things happen, binding together all the objects. Moreover, specific subprograms in each product component help us to interact with things outside the standard Oracle 8*i* and Developer /2000 environment.

FEATURES OF THE SOFTWARE

In the present IT environment, the RDBMS concept has gained immense importance. This is due to several factors. They can be listed as follows:

RECOVERY

Recovery is the ability of the database manager to recover the database after transaction, application, system and media failures. If we write an application that depends on a file system and direct file access, we totally expose ourselves to any failure at any level. Our software has to take such failures into account. Usually it deals with transaction failures, but other kinds of failure often damage the underlying data.

Between Developer /2000 and the underlying database manager, there is nothing to worry as it is occasionally required for the database manager to restore a damaged database file somewhere.

The transaction begins with a data-related action and ends with a successful termination command *(COMMIT)* or an unsuccessful termination command *(ROLLBACK)*.

If the *error is not fatal*, the transaction rolls back and that is the end of it. If the *error is fatal* to the application, such as an invalid entry into a field, the database manager rolls back the current transaction if the error is

DATA INDEPENDENCE

One of the central reasons for database management is the need for a central database to support multiple applications, as well as multiple users of one application. The *internal model* is the physical layout of the data: the access method structure, B-trees, files structures, secondary indexes, and so on. The *conceptual model* is the centralized, logical data model: the tables with their rows and columns.

System design and Development

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 INPUT DESIGN

Input design is a part of overall system it is the phase that requires careful attention and is the most important one Objective during input design are as follows:

a. Achieve high level accuracy
b. Ensure input is free of ambiguity

The input design involves converting the user-originated inputs into a computer-based format. The aim of input design is to make data entry easier, logical error free. It helps us to filter errors in the input data that otherwise entered into the database might have brought in a lot of consistency.

Involves procedures for capturing data, verifying and then passing them on to system. After choosing input medium, attention is focused designing of error handling, control, and grouping and validation procedures.

During application development, care has been taken to make our system extremely user-friendly and organize our screens such that the possibilities of making error are maintained.

List of possible values are list of value (LOV) , display item , Alert, popup menu. This makes system less error prone to errors, as the input is selected rather than typing.

Warnings for wrong entries such as program number with a character, negative value. Each and every field entered by user verified with appropriate validation procedure.

INPUT TYPES

One of the early activities of input design is to determine the nature of input data. The different types of inputs handled by our system are:

External	:	Prime input to system
Internal	:	Communication with system
Operational	:	Programming team communicating with system.
Computerized	:	Input to the computer media from other internal sources.
Interactive	:	Input entered during a dialog.

'Commitment Monitoring System' does not have any menu. Because users wanted very friendly environment, so user interfaces are define as 'Tool bar' and buttons. Input design uses the text item, display item, list item, radio button to collect the values from the user. Data should be entered for all the fields. All the data are validated and if there is an invalid data a message box is flashed in the screen. This enables the user to interact with the application.

4.2 OUTPUT DESIGN

Output is the hart of the all software design. The developed system will be said successful only if the output system provides the necessary reports in necessary format. The other important factor related with output design is user's estimation. Every user of the system estimates it only based on the given by the system, so to keep our software in high grade we need to compose an efficient output design.

Characteristics of the output design

- Fit for user needs – The output will produce what the user needs provides outputs on time outputs should be produce at the time when it is needed if it is not then the information will not be useful for decision making.
- The outputs should abstract the complexity of the system
- The output should be accurate and easy to understand
- Output should be properly formatted
- Output design should support for making hard copies and backups

Fincap status

When particular authority entered in to an approval module a separate option is there for viewing the status of fincaps to be approved.

Also a query option is allowed to view the particular fincap or purchase order.

Overall status

When administrator entered his valid user id and password then the overall status will list the authority level fincaps and findisposal to be approved. Ther is another option to view all the approved fincaps. Here you have to enter the from and to date to view the approved fincaps.

.

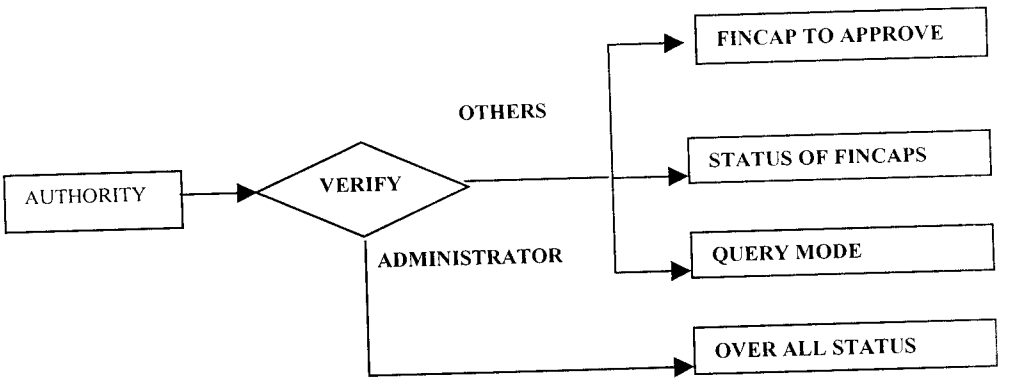


Fig .1 approval status

Following reports are available in this system,

- Pending fincaps with out detail
- Pending fincaps with detail
- Approved fincaps
- Supplier wise detail
- Program wise detail
- Purchase order detail
- Findisposal detail

4.3 DATABASE DESIGN

DATABASE MANAGEMENT SYSTEM CONCEPTS

Introduction

Data are raw facts that we use to represent information. Processed Data is information. Data must be manipulated (Organized, Formatted, Summarized, etc.) before it can be used as information.

Database is used to store data in an organized fashion, which allows us to access and manipulated the data. The techniques and data structures used to organize and manipulate data in databases are collectively known as **Data Modeling.**

Data Base management Systems (DBMS) serve to manipulate and maintain databases. When industry's need for information was small, database tended to be simple and informal. But as the need for up-to-date information increased, automated DBMS were developed based on groups of formalized data modeling rules called **Data Models.**

Three data models that are available are:

- Hierarchical
- Network (CODASYL)
- Relational

Hierarchical Database Model

The hierarchical database model is the oldest approach to databases. It evolved from sequential file structures in which little effort was to isolate the logical data structure from the physical data structures on the storage

Network Data Model

The Network data model is similar to the hierarchical data model, except that its records are not limited to only one *superior*. A record may have many *superior* records and many *subordinates* that are linked to the record's *superior*.

Relational Database Model

The relational data model was first described by **E.F codd** in 1970; one year after the **Data Base Task Group (DBTG)** published its first proposal. In the relational data model, entity types are referred to as relations. The relational model was an attempt to simplify Database structure. It represents all data in the database as simple row-column tables of data values and where all database operations work on these tables.

Relations

A relation is a formal term for a table. A relational database is defined as a collection of tables called *Relations*. In relational terms, a record (table row) is called *Tuple,* and the fields (columns) are called *Attributes*.

The number of tuples is called the Cardinality and the number of attributes is called the *Degree*. Every table must have some column or combination of columns that uniquely identify each row in the table.

This column (or columns) is called the *Primary Key* of the table. A Domain is a pool of values from where one or more attributes (columns) draw out their actual values. In relational systems, missing or unknown information can be represented as *Null*.

Relational Operators

The relational model is based on the solid foundation of relational algebra. Relational algebra consists of a collection of operators that operate on relations. Each operator takes one or two relations as its input and produces a new relation as its output. The main objective of relational algebra is data retrieval. The relational operators are Project, Cartesian Product, union, Intersect, Difference and Join.

Relationship

Relationships refer to the mapping of Relations. The various types of relationships are

- One-to-One
- One-to-Many
- Many-to-One
- Many-to-Many

Normal Forms

Normalization results in the formation of tables that satisfy certain specified constraints, and represents certain Normal forms

Several normal forms have been identified. The most important and widely used of these are.

- First Normal Form (1NF).
- Second Normal Form (2NF).
- Third Normal Form (3NF).
- Boyce-codd Normal Forms (BCNF).

Functional Dependency

Given that A and B be composite attributes and R is a relation, attribute A is functionally dependent on B, if each Value of A in R is associated with precisely one value of B.

First Normal Form

This is the lowest level of normalization. It states that data is in First Normal Form (1NF) if the pool of valid values that may appear in an attribute contains only atomic values. (Atomic values cannot be decomposed into smaller units) Each column can contain only one value in any row of a table.

Second Normal Form (2NF)

Data is Second Normal Form (2NF) if it is in 1NF and every attribute in the record is functionally dependent upon the whole key and not just a part of the key.(an attribute is a non Key if it is not part of the primary key). The purpose of 2NF is to eliminate repeating groups and to ensure that the remaining attributes belong to this entity. An attribute is functionally dependent on a key if the attribute contains only one value which depends on the key.

Third Normal Form (3NF)

Data is Third Normal Form (3NF) if and only if it is in 2NF and every non-key attribute is non transitively dependent on the primary key. The purpose of 3NF is to ensure that the attribute directly belongs to the entity.

Other Normal Forms

The other forms are Boyce Codd Normal Forms (BCNF). Fourth Normal Form (4NF) and Fifth Normal Form (5NF). They are seldom used.

DATA DICTIONARY

Tables store two types of information in the database user data and system data. Application will access user data most often, this is data that is inserted, updated or deleted through the database applications.

Oracle needs to maintain large amount of system data, which describes the database system and is used by the oracle during its operations in systems tables- I.C. tables that are created and managed by oracle.

System data is used to manage the database. It contains description of all the objects in the system, including the tables, columns, views, indexes and synonyms. The set of tables that oracle uses for the system uses of the system data is called Data Dictionary.

TABLE NAME: PROGRAM MASTER

COLUMN NAME	DATA TYPE & SIZE	CONSTRAINT	DESCRIPTION
Program_No	Number(3)	Primary key	Program Number
Program_Name	Varchar2(50)	Not null	Program_Name
Program_Date	Date		Program Date
Total_Approval	Number(14,2)	Not null	Total Approval
Cleared_For_Commitment	Number(14,2)	Not null	Cleared For Commitment
Commitment_Ytd		Not null	Commitment Year To Date
Mpb_Fc	Number(14,2)		Master Plan Forecast
Capex_From_Start	Number(14,2)		Capex From Start
Capex_Ytd	Number(14,2)		Capex Year To Date
Fincap_No_Change_Flag	Char(1)		Fincap No Change Flag
Fincap_No	Number(3)	Not null	Fincap Number

TABLE NAME: PURCHASE ORDER MASTER

COLUMN NAME	TYPE & SIZE	CONSTRAINT	DESCRIPTION
Program_No	Number(3)	Foreign kye	Program Number
Fincap_No	Number(3)	Foreign kye	Fincap Number
Po_No	Number(6)	Primary key	Purchase Order Number
Po_Date	Date		Purchase Order Date
Supplier_Code	Varchar2(4)	Not null	Supplier Code
Supplier_Name	Varchar2(50)		Supplier Name
Po_Category	Varchar2(20)		Purchase Order Category
Po_Type	Varchar2(10)		Purchase Order Type
Terms	Varchar2(200)		Terms
Attach_Path_Address	Varchar2(50)		Attach Path Address
Attach_Path_Address2	Varchar2(50)		Attach Path Address2
Modification_No	Number(5)	Not null	Modification No

TABLE NAME: PURCHASE ORDER DETAIL

COLUMN NAME	TYPE & SIZE	CONSTRAINT	DESCRIPTION
Program_No	Number(3)	Foreign key	
Fincap_No	Number(3)	Foreign key	Fincap_No
Po_No	Number(6)	Foreign key	Po_No
Description	Varchar2(100)		Description
Quantity	Number(4)		Quantity
Units	Varchar2(5)		Units
Rate	Number(20,2)		Rate
Item_Value	Number(20,2)	Not null	Item_Value

TABLE NAME: MODIFICATION MASTER

COLUMN NAME	DATA TYPE & SIZE	CONSTRAINT	DESCRIPTION
Oldpo_No	Number(6)	Not null	Old Purchase Order Number
Modification_No	Number(6)	Primary key	Modification_Number
Modification_Date	Date		Modification_Date
Supplier_Code	Varchar2(4)	Not null	Supplier Code
Supplier_Name	Varchar2(50)		Supplier Name
Po_Category	Varchar2(20)		Purchase Category
Po_Type	Varchar2(10)		Purchase Order Type
Terms	Varchar2(200)		Terms
Attach_Path_Address	Varchar2(50)		Attach Path Address
Attach_Path_Address2	Varchar2(50)		Attach Path Address2
Fincap_No	Number(3)	Foreign key	Fincap Number
Program_No	Number(3)	Foreign key	Program Number
Oldfincap_No	Number(3)	Not null	Old Fincap Number

TABLE NAME: MODIFICATION DETAIL

COLUMN NAME	TYPE & SIZE	CONSTRAINT	DESCRIPTION
Modification_No	Number(6)	Foreign key	Modification No
Description	Varchar2(100)		Description
Quantity	Number(4)		Quantity
Units	Varchar2(5)		Units
Rate	Number(14,2)		Rate
Item_Value	Number(14,2)		Item_Value

TABLE NAME: SUPPLIER MASTER

COLUMN NAME	TYPE & SIZE	CONSTRAINT	DESCRIPTION
Supplier_code	Varchar2(4)	NOT NULL	Supplier code
Supplier_name	Varchar2(50)		Supplier name
Address	Varchar2(25)		Address
Address1	Varchar2(25)		Address1
Address2	Varchar2(25)		Address2
Pincode	Number(6)		Pincode
State	Varchar2(20)		State
Contact_person	Varchar2(25)		Contact person
Phone1	Varchar2(15)		Phone1
Phone2	Varchar2(15)		Phone2
Fax	Varchar2(15)		Fax
Mobile	Varchar2(10)		Mobile
Email	Varchar2(50)		Email

4.4 PROCESS DESIGN

The Commitment Monitoring System has the following processing systems. They are

- Program processing
- Order processing
- Fincap processing
- Approval processing

Program processing

Program processing deals with the creation of the program number every year and project planning. For which a value limit will be decided and that amount of value will be used to put orders. In project planning the program number does not change every year.

Order processing

In order processing two types of orders will be put and the order value will be added in the commitment ytd of program master. Once the order exceeds the cleared for commitment. Then cannot put an order further for that particular program number. Many details are maintained in the order processing which is necessary to put an order.

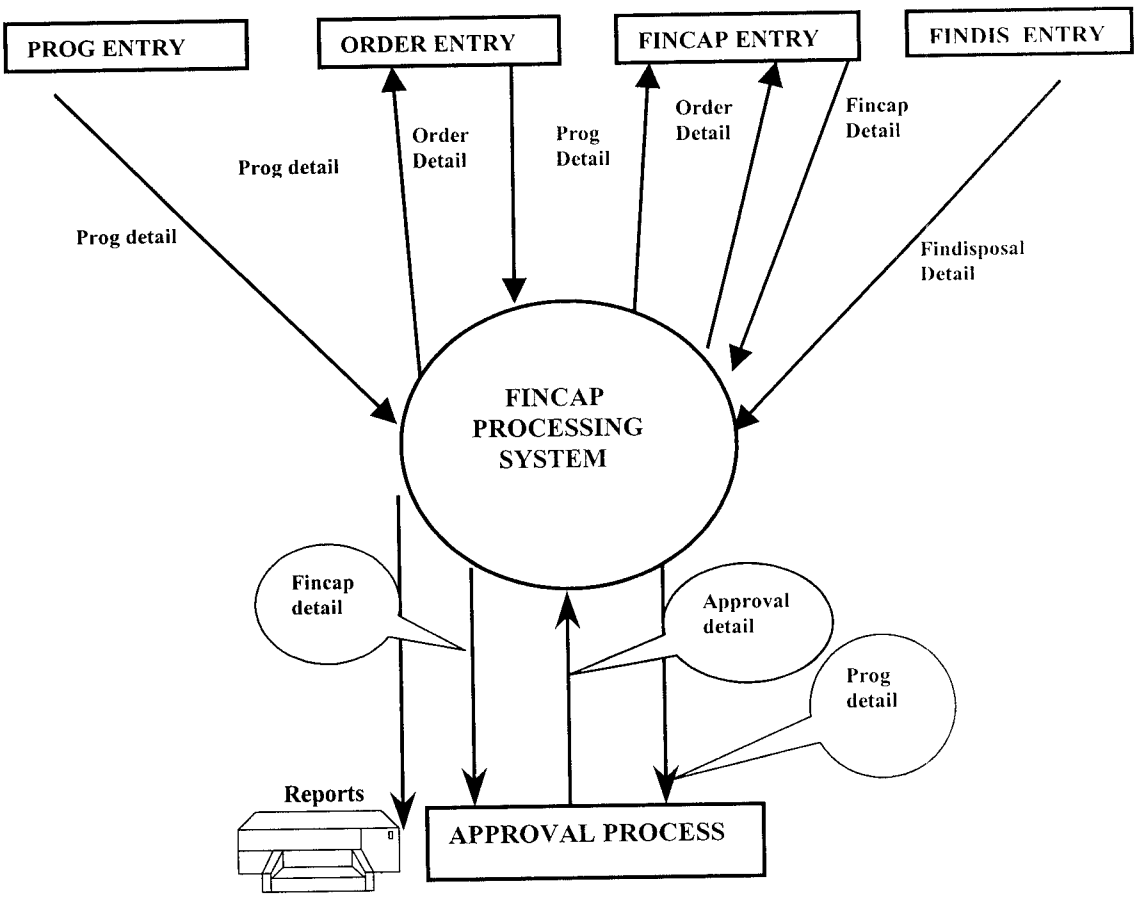
Fincap processing

In the fincap processing a fincap will be processed for one or more order. Order may be any one of the following category Machine tools, T&GSE, Services, Others and Civil. In this process previous release, release now, total amount, balance is calculated.

Approval processing

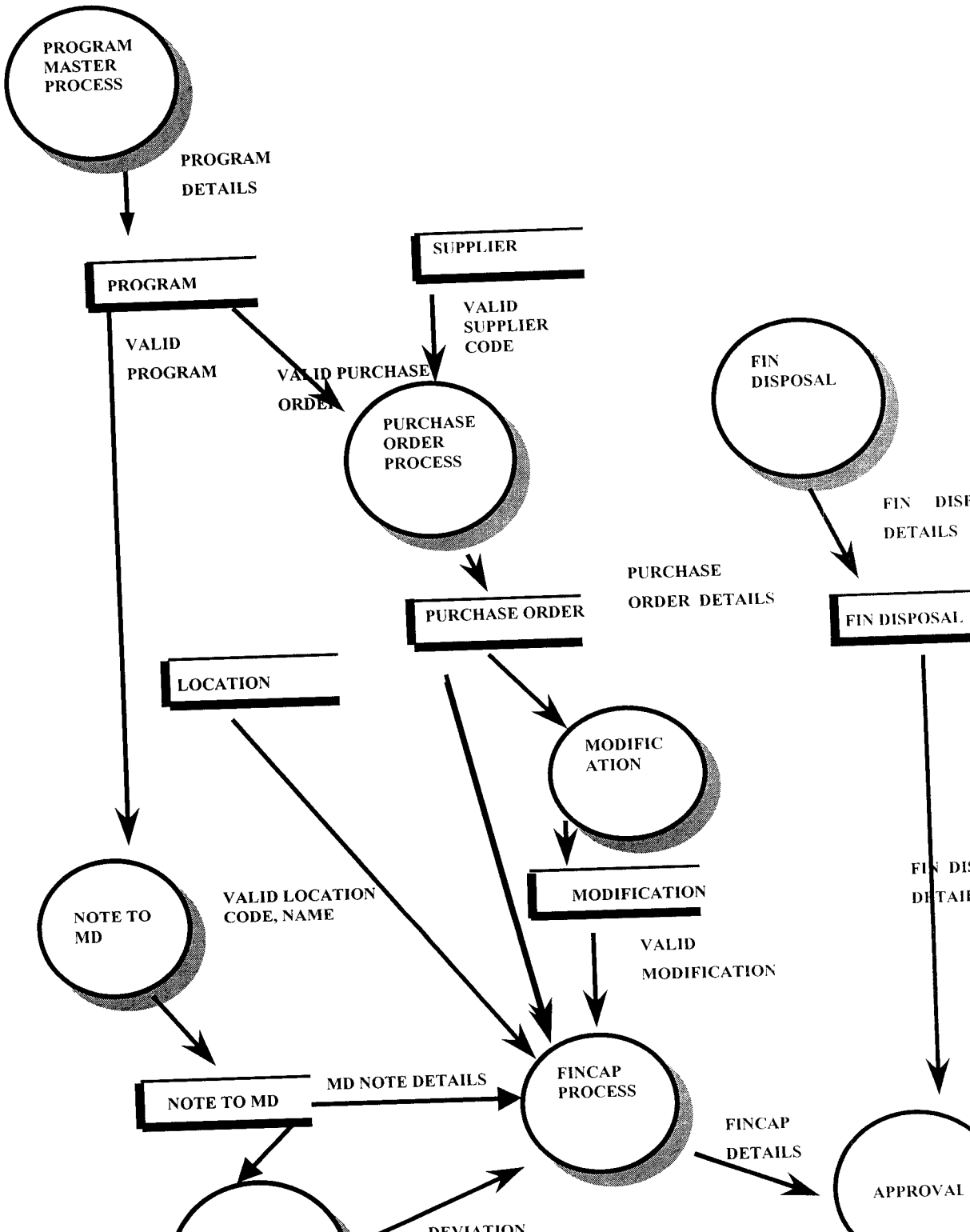
In approval processing the fincap will be viewed by many divisions. They view all the order details, program, fincap details and authorize it and sent it to the next level of authorization.

CONTEXT ANALYSIS DIAGRAM



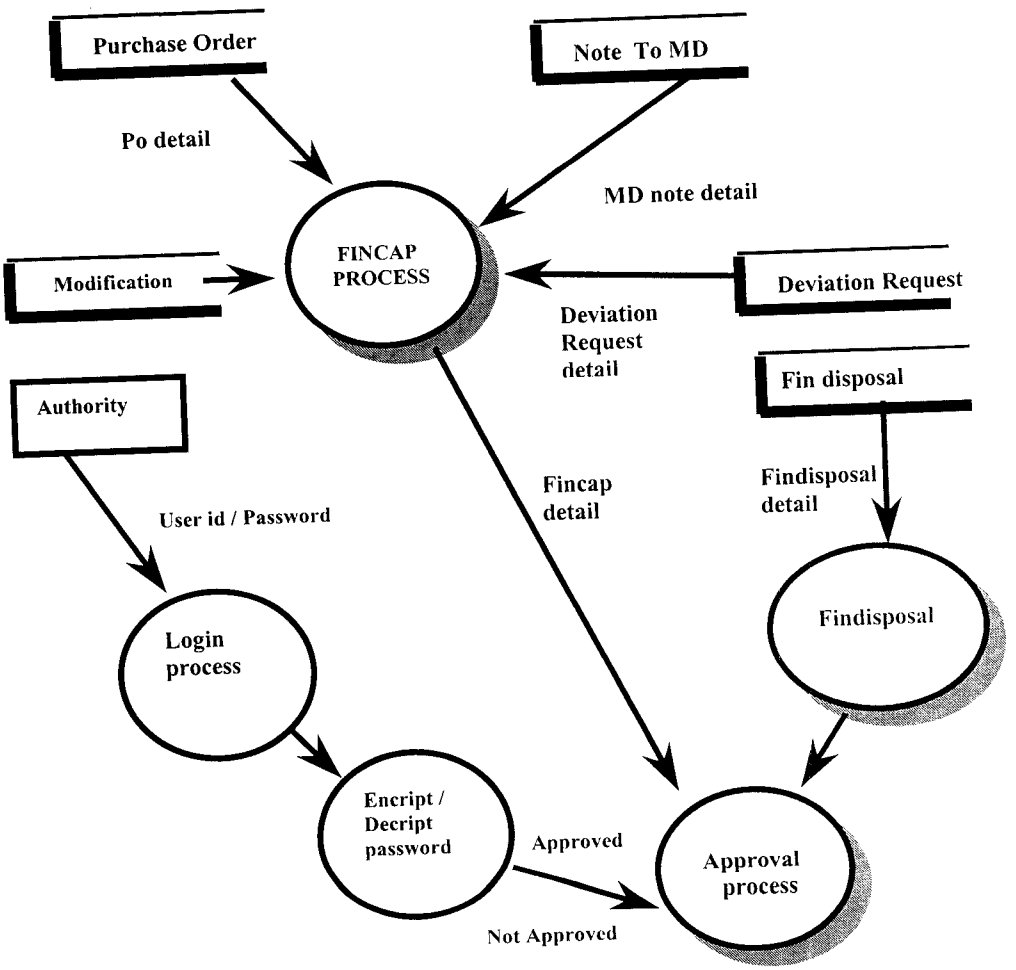
OVERALL PROCESSING OF CMS

LEVEL 1 DFD



APPROVAL PROCESS

LEVEL 2 DFD



System Testing and Implementation

5.2 SYSTEM TESTING

Software testing is an important phase in the development of the software. Testing is done for software quality assurance. Software is developed for a long use in the future so it is necessary that a well planned testing be done to overcome errors. Static Analysis is used to investigate the structural properties of the source code. Dynamic test cases are used to investigate the behavior of source code by executing the program on the test data. As before, we use the term "Program Unit" to denote a routing or a collection of routines or a routine implemented by an individual programmer. In a well-designed system, a program unit is stand-alone or a functional of a large system.

UNIT TESTING

Unit testing comprises the set of tests performed by an individual programmer prior to integration of the unit into a larger system. The situation is illustrated as follows,

Coding&debugging ------ > Unit testing ------ > Integration testing

A program unit is usually small that the programmer who developed it can test it in great detail and that will be possible when the unit is integrated into a evolving software product. There are four categories of test a programmer will typically perform on a program unit.

- Functional test
- Performance test
- Stress test
- Structure test

Functional test cases involve exercising the code with nominal input values for which the expected results are known, as well as boundary values like minimum values, and values on and just the functional boundaries and special values, such as logically related inputs, the identity matrix, files of identical elements, and empty files.

Performance test determines the amount of execution time spent in various parts of the unit, program throughput, response time and device utilization by the program unit. A certain amount of performance tuning may be done during unit testing; however, caution must be exercised to avoid expending too much of fine tuning of a program unit that contributes little to the overall performance of the entire system. Performance testing is most productive at the subsystem and system levels.

Stress test is those tests designed to intentionally break the unit. A great deal can be learned about the strengths and limitations of the program by examining the manner in which a program unit breaks.

Structure tests are concerned with exercising the internal logic of the program and traversing particular execution paths. The functional, performance and stress testing are called as "Black Box" testing.

The main activities in structural testing are deciding which paths to exercise, deriving test data to exercise those paths, determining the test coverage criterion to be used, executing the test cases and measuring the test coverage achieved when the test cases are exercised.

A test coverage criterion must be established for unit testing, because program units usually contain too many paths to permit exhaustive testing. Even if it ere possible to successfully test all paths through a program, correctness would not be guaranteed by path testing because the program might have missing paths and computational errors that were not discovered by the particular test cases chosen. A missing path error occurs when the branching statement and the associated computations are accidentally omitted. Missing path errors can only be detected by functional test cases from the requirement specifications. Thus, tests based solely on the program structure cannot detect potential errors in a source program.

Unit testing focuses on verification. In the smallest unit of software design the module unit testing is done for each module to ensure that it functions properly as a unit. In unit testing, the module interface is tested to ensure that information properly flows into and out of the program under test.

Unit testing is done to recover errors of the following types.

- Erroneous initialization
- Incorrect variable names
- Inconsistent data type
- Underflow, overflow and addressing exceptions
- Computation errors.

SYSTEM TESTING

System testing is a series of different tests whose purpose is to exercise the computer-based system fully. There are two kinds of activities viz. Integration testing and Acceptance testing. Careful planning and scheduling are required to ensure the modules will be available for integration into the evolving software product when needed. Acceptance testing involves planning and execution of functional tests, performance tests and stress tests to verify that the implemented system satisfies its requirements. Acceptance tests are typically performed by the

circumstances, the development group may or may not be involved in the acceptance testing.

INTEGRATION TESTING

Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to recover errors associated with interface. Bottom up integration is the traditional strategy to integrate the components of the software system into a functional as a whole. Bottom up integration consists of unit testing, followed by subsystem testing, and followed by testing of the entire system. Unit testing has the goal of discovering errors in the individual modules of the system. Modules are tested in isolation from one another in an artificial environment known as "Test Harness" which consists of the driver programs and data necessary to exercise the modules.

Unit testing should be as exhaustive as possible to ensure that each representative case handled by each module has been tested. Unit testing is eased by a system structure that is composed of small, loosely coupled modules.

A subsystem consists of several modules that communicate with each other through well-defined interfaces. Normally, a subsystem implements a major segment of the total system. The primary purpose of subsystem testing is to verify operation of the interfaces between modules in the subsystem. Both control and data interfaces must be tested. Large software systems may require several levels of subsystem testing: Lower-level subsystems are successively combined to form higher-level subsystems. In most software systems, exhaustive testing of subsystem capabilities is not feasible due to the combination complexity of the module interfaces; therefore test cases must be carefully chosen to exercise the interfaces in the desired manner.

System testing is concerned with subtleties in the interfaces, decision logic, control flow, recovery procedures, throughput, capacity and timing characteristics of the entire system. Careful test planning is required to determine

Disadvantages of bottom up testing include the necessity to write and debug test harnesses for the modules and subsystems, and the level of complexity that result from combining modules and subsystems into larger units. All the modules are then linked and execution in a single integration run. This is the "Big Bang" approach to integration testing.

The main problem with big-bang integration is the difficulty of isolating the sources of errors.

Top down integration testing starts with the main routine and one or two immediately subordinate routines in the system structure. After this top- level "Skeleton" has been thoroughly tested, it becomes the test harness for its immediately subordinate routines. Top down integration requires the use of program stubs to simulate the effort of lower level routines that are called by those being tested. Disadvantages of top down integration are,

- System integration is distributed throughout the implementation phase. Modules are integrated as they are developed.
- Top-level interfaces are tested first and most often.
- The top-level routines provide a natural test harness for lower-level routines.
- Errors are localized to the new modules and interfaces that are being added.

While it may appear that top-down integration is always preferable, in many situations it is not possible to adhere to a strict top-down coding and integration strategy. For example it may be difficult to find top-level input data that will exercise a lower level module in a particular desired manner.

Sandwich integration is predominantly top-down, but bottom-up techniques are used on some modules and subsystems. This mix alleviates many of the problems encountered in pure top-down testing and retains the advantage of top-down integration at the subsystem and system level.

ACCEPTANCE TESTING

Acceptance testing involves planning and execution of functional tests, performance tests and stress tests in order to demonstrate that the implemented system satisfies its requirements. It is not unusual for two sets of acceptance tests to be run: those developed by quality assurance group and those developed by the customer.

In additional to functional and performance tests, stress tests are performed to determine the limitations of the system. Typically, acceptance tests will incorporate test cases developed during unit testing and integration testing. Additional test cases are added to achieve the desired level of functional, performance and stress testing of the entire system.

5.3 PERFORMANCE

- Provides quick and timely information.
- Reduces unnecessary complaints.
- Reduces the documents processing time/turnaround time of documents.
- Greater accuracy.
- User friendly
- Reduces costs.
- Reduces working time.
- Immediate references.
- Any number of records can be easily retrieved.
- Multiple copies of reports can be easily taken.
- Appropriate error messages are displayed.

Commitment Monitoring System, which is developed in developer 2000 with oracle, is a powerful application. The software is hassle free software that simplifies the job of the various departments.

The following are the major performance of the software.

- Faster access of the data while retrieving the data
- Simplifies the work of the user
- Proper maintenance of data
- Faster in generating reports
- Good user interface facility
- Bugs free software
- Good in giving appropriate messages where ever need to help the user
- Maintains data integrity

5.4 Refinements Based on Feedback

This application is provided with two set of feedbacks. They are

(i) Operator feedback
(ii) Online feedback

The feedbacks are received from the users of the system. The feedback is used to get suggestions or comments about the system. The feedbacks provided by the users are of great importance. From the feedback we may be able to get valuable suggestions from the users. The suggestions may be implemented in the system. Observing users executing the tasks and collecting data and feedback. Feedback is organized, analyzed, and interpreted. These process helps to identify the problems which are encountered by the users, giving the reasons for the problems and offering possible solutions by rectifying the problem.

The feedback that we received from the users of the application are very encouraging and appreciative ,their suggestions were made worthwhile and are seriously considered.

Future Enhancements

6. FUTURE ENHANCEMENT

Any Software that is developed cannot be said to be perfect since computer science is a very dynamic field. Therefore changes are unavoidable. Nothing can be said to be perfect at this age and time where every day scientific miracles take place. This software development project deals with Sales Order, Purchase Order, Issue process of stocks, and maintaining the current stock reports.

'Commitment Monitoring System' processing system is a vast concept, which can be enhanced by including the computerization of modification of the existing Fincaps, as this concept deals with controlling and coordinating all the transaction process within the organization.

Conclusion

7. CONCLUSION

I would like to conclude that this project **'Commitment Monitoring *System*** ' has been selected and developed by keeping in mind the various drawbacks of the manual system presently followed in majority of the organizations and the various problems faced by the companies following the computerized method. Their requirements have been collected and based on such details; an effort has been made to cater to the needs of all the sectors.

This system is presently being implemented in Ashok Leyland Ltd, Corporate office in Chennai.

This system will be the Organization in taking appropriate decisions and study the problems. This module satisfies the special requests by the departments.

Hence I conclude that proposed computerized system would function and meet the requirements.

References

REFERENCES

Books:

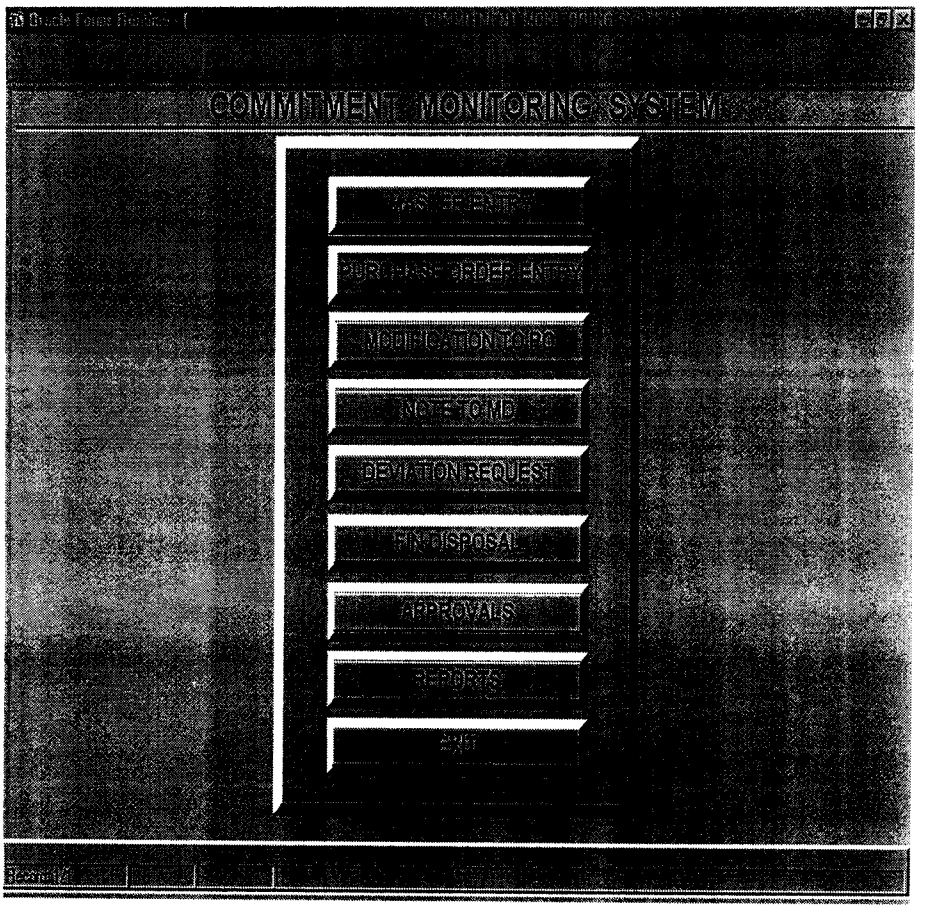
1. "Oracle Complete Reference" Oracle Corporation 1999
2. "Building Reports Manual " Oracle Corporation 1999
3. "SQL & PL/SQL Reference Manual" Oracle Corporation 1998
4. Ivan N. Bayross "Commercial Application Development Using Oracle Developer 2000" BPB Publications 2rd Edition 2000

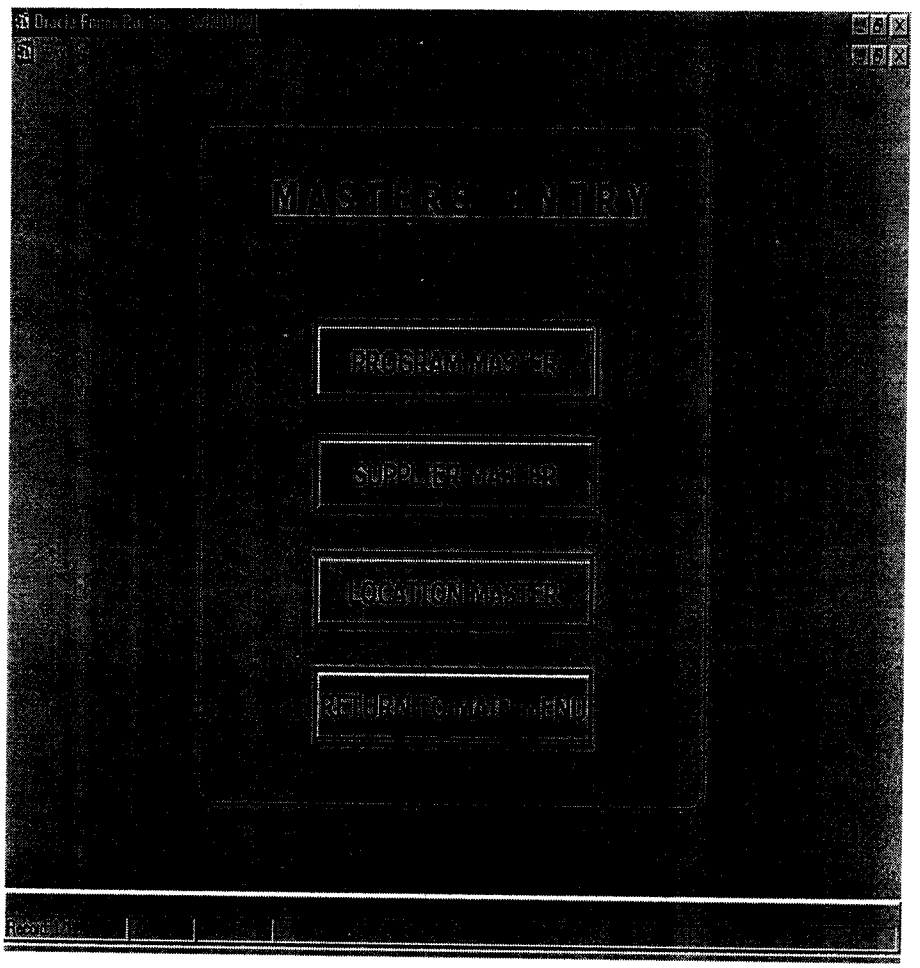
Website:

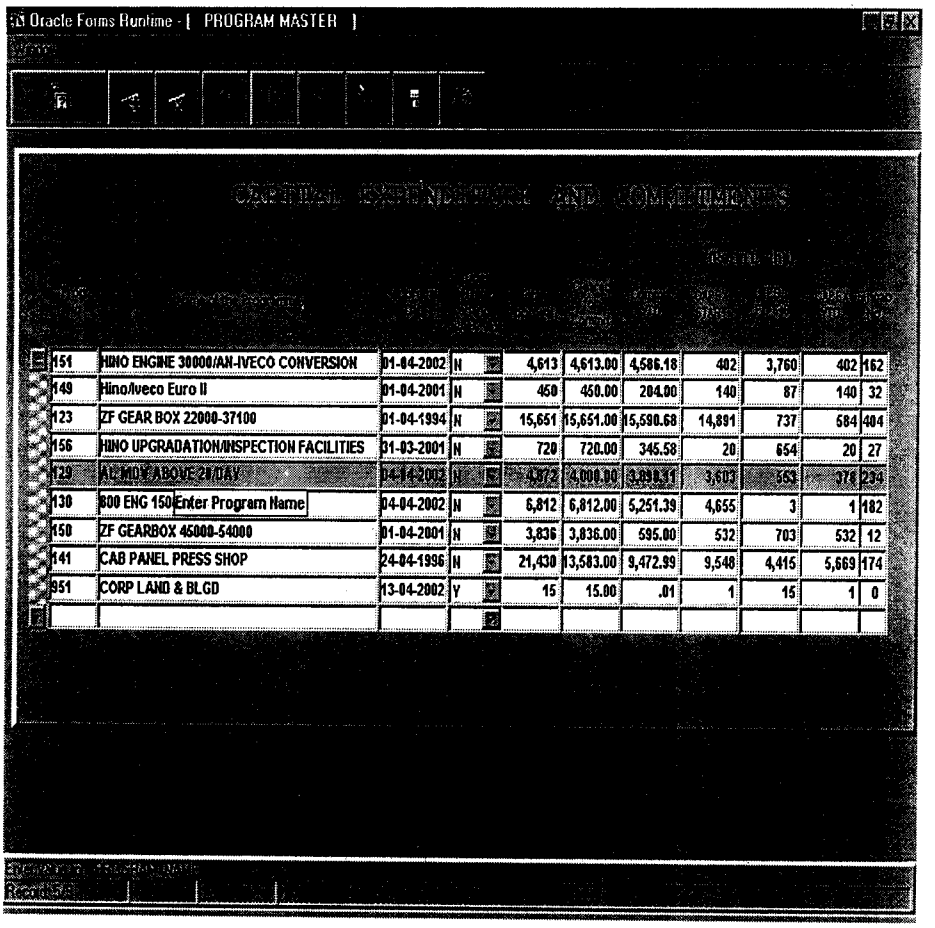
www.oracle.com

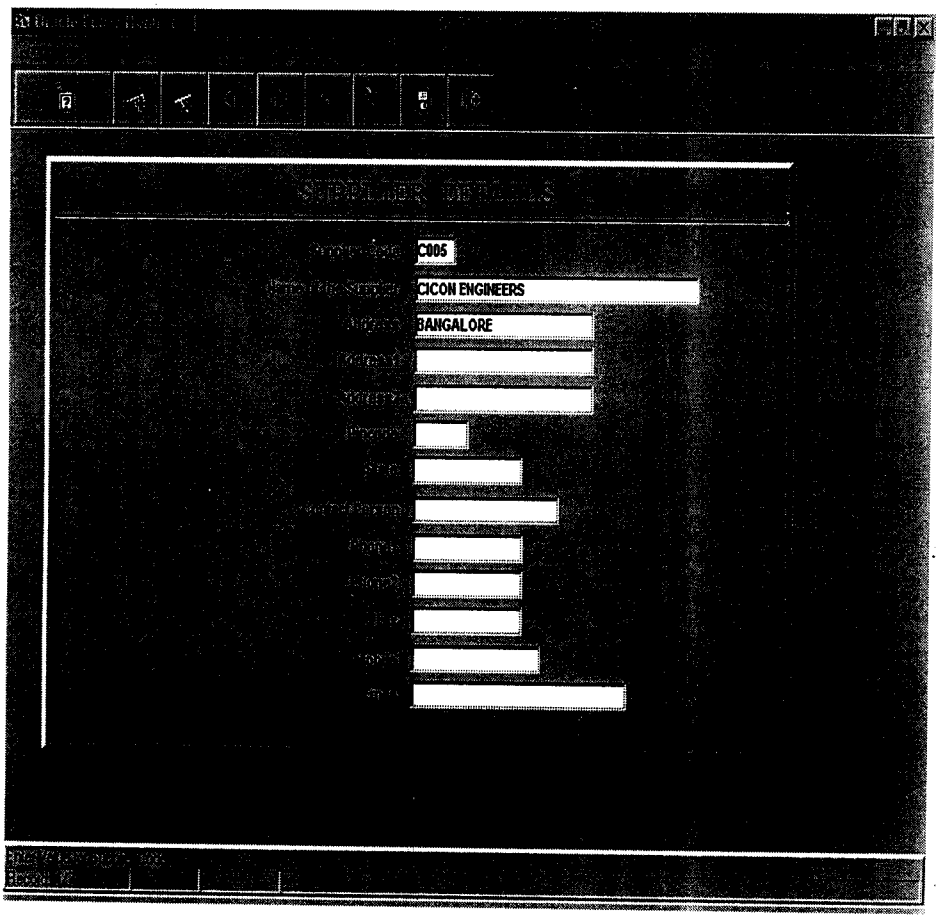
Appendices

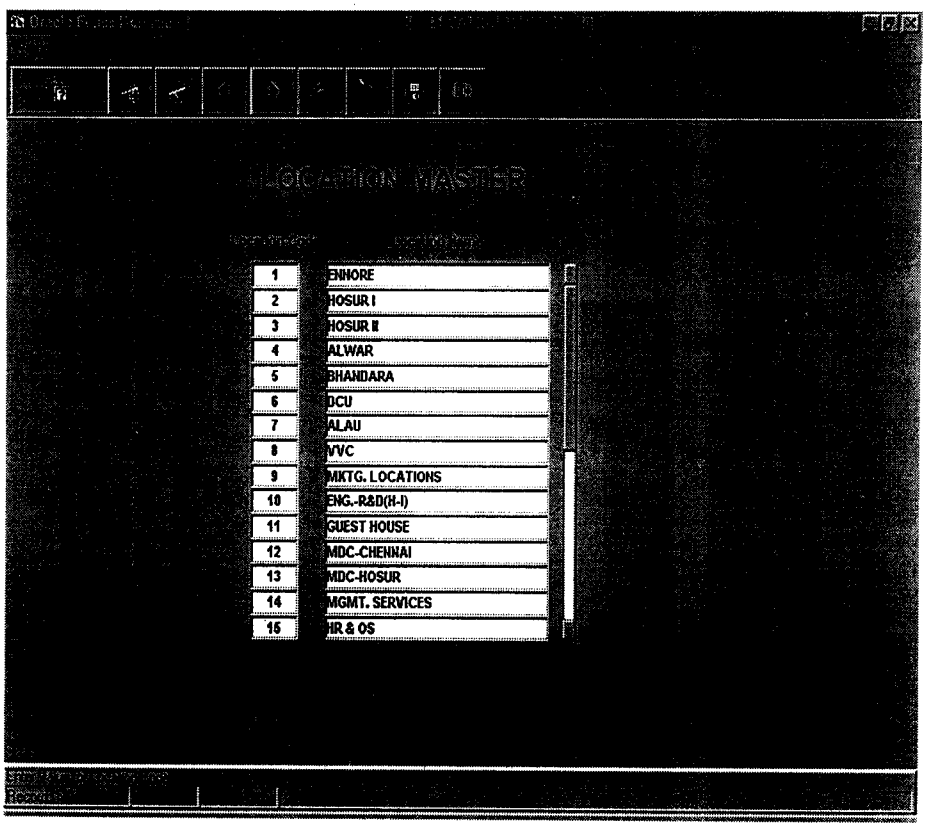
SAMPLE SCREENS

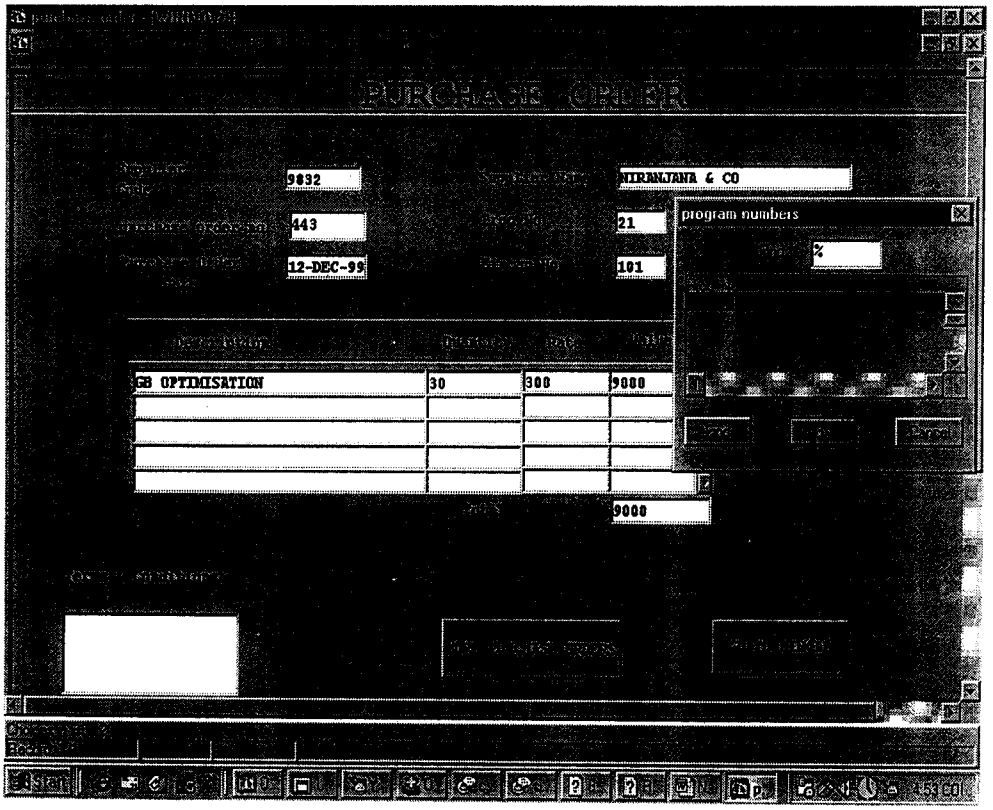

COMMITMENT MONITORING SYSTEM
MODIFICATION TO PO
NOTE TO MD
DEVIATION REQUEST
FIN DISPOSAL
APPROVALS
REPORTS
EXIT

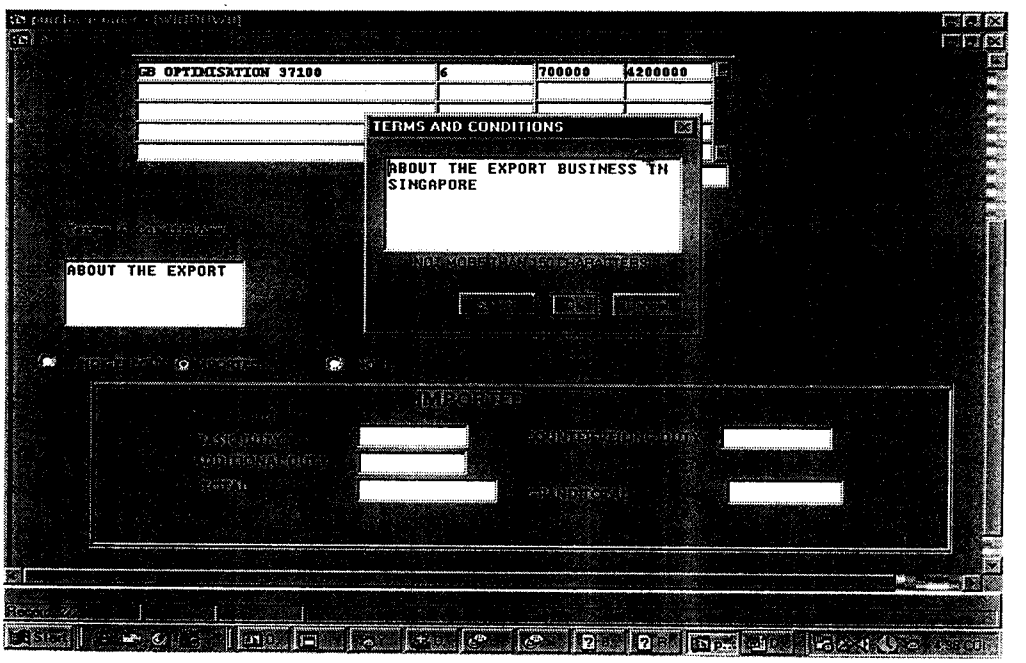

MASTER ENTRY

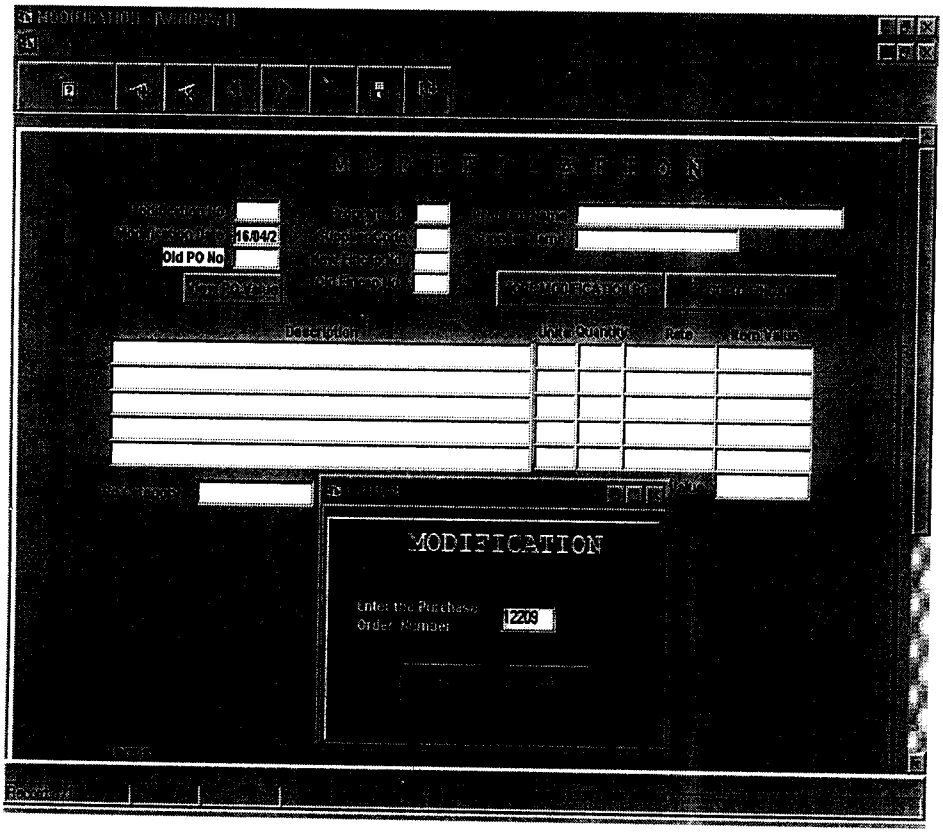

Oracle Forms Runtime - [PROGRAM MASTER]
151 HINO ENGINE 30000/AN-IVECO CONVERSION 01-04-2002 N 4,613 4,613.00 4,586.18 402 3,760 402 162
149 Hino/Iveco Euro II 01-04-2001 N 450 450.00 284.00 140 87 140 32
123 ZF GEAR BOX 22000-37100 01-04-1994 N 15,651 15,651.00 15,590.68 14,891 737 584 404
156 HINO UPGRADATION/INSPECTION FACILITIES 01-03-2001 N 720 720.00 345.58 20 654 20 27
129 ACMUX ABOVE 2M/DAY 04-04-2002 N 4972 4,000.00 3,393.31 3,603 553 378 234
130 600 ENG 150 Enter Program Name 04-04-2002 N 6,812 6,812.00 5,251.39 4,655 3 1 182
150 ZF GEARBOX 45000-54000 01-04-2001 N 3,836 3,836.00 595.00 532 703 532 12
141 CAB PANEL PRESS SHOP 24-04-1996 N 21,430 13,583.00 9,472.99 9,548 4,415 5,669 174
951 CORP LAND & BLGD 13-04-2002 Y 15 15.00 .01 1 15 1 0

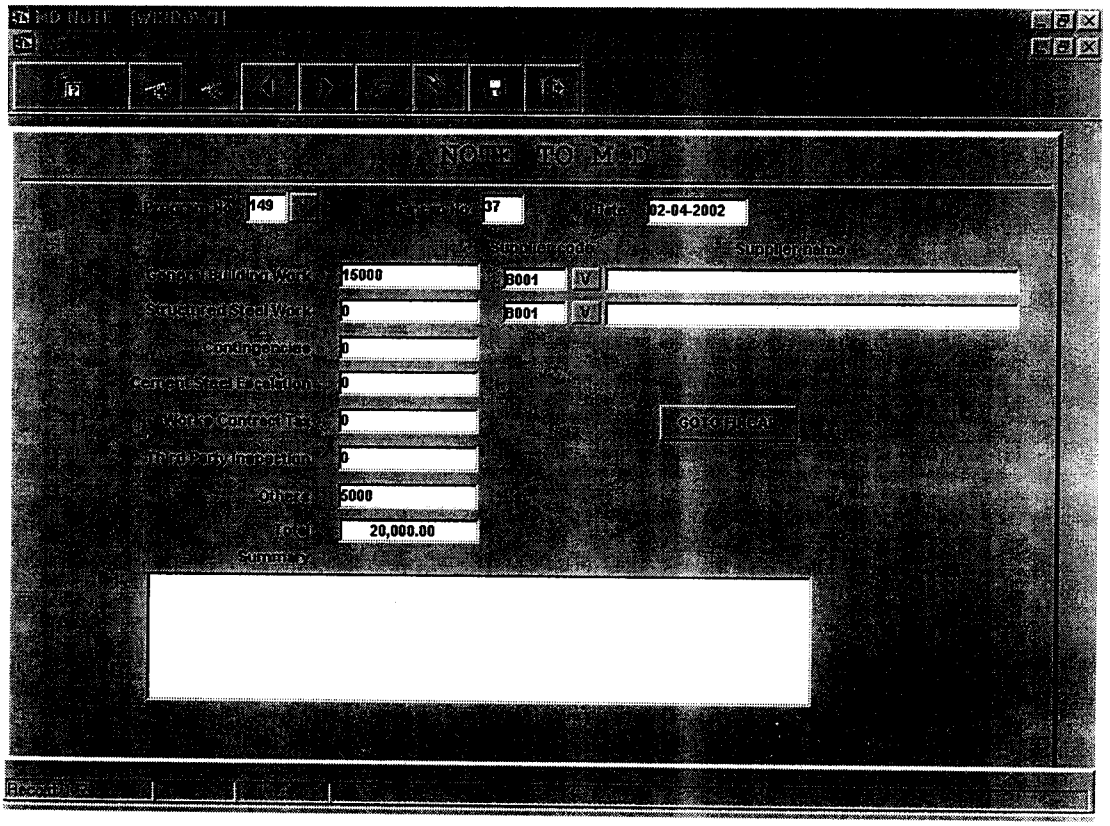

C005
CICON ENGINEERS
BANGALORE

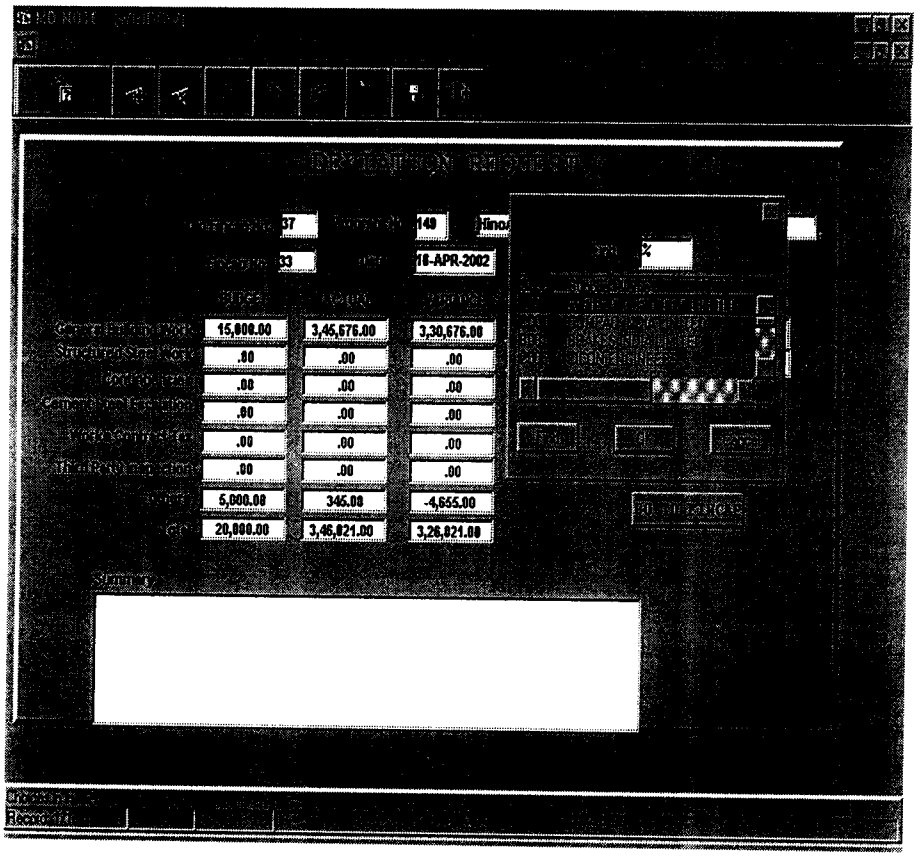

LOCATION MASTER
1 ENNORE
2 HOSUR I
3 HOSUR II
4 ALWAR
5 BHANDARA
6 DCU
7 ALAU
8 VVC
9 MKTG. LOCATIONS
10 ENG.-R&D(H-I)
11 GUEST HOUSE
12 MDC-CHENNAI
13 MDC-HOSUR
14 MGMT. SERVICES
15 HR & OS

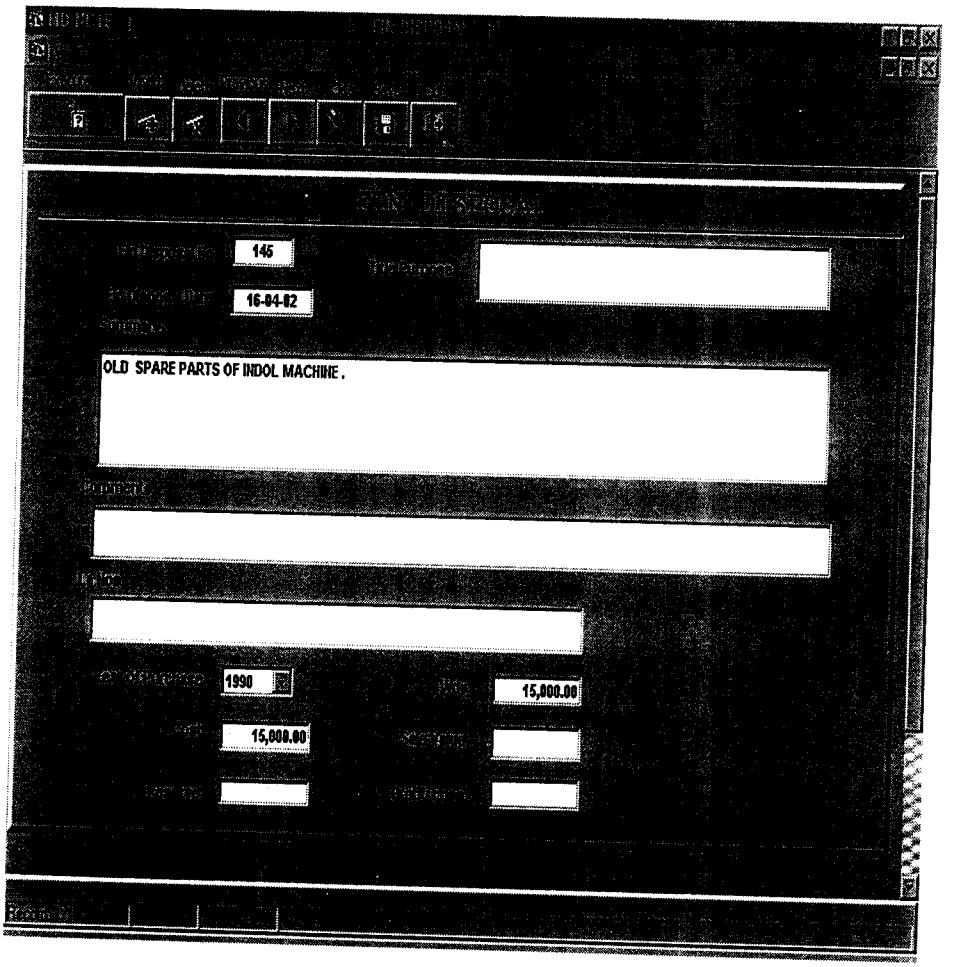

PURCHASE ORDER
9832
443
12-DEC-95
NIRANJANA & CO
21
101
program numbers
%
GB OPTIMISATION
30
300
9000
9000

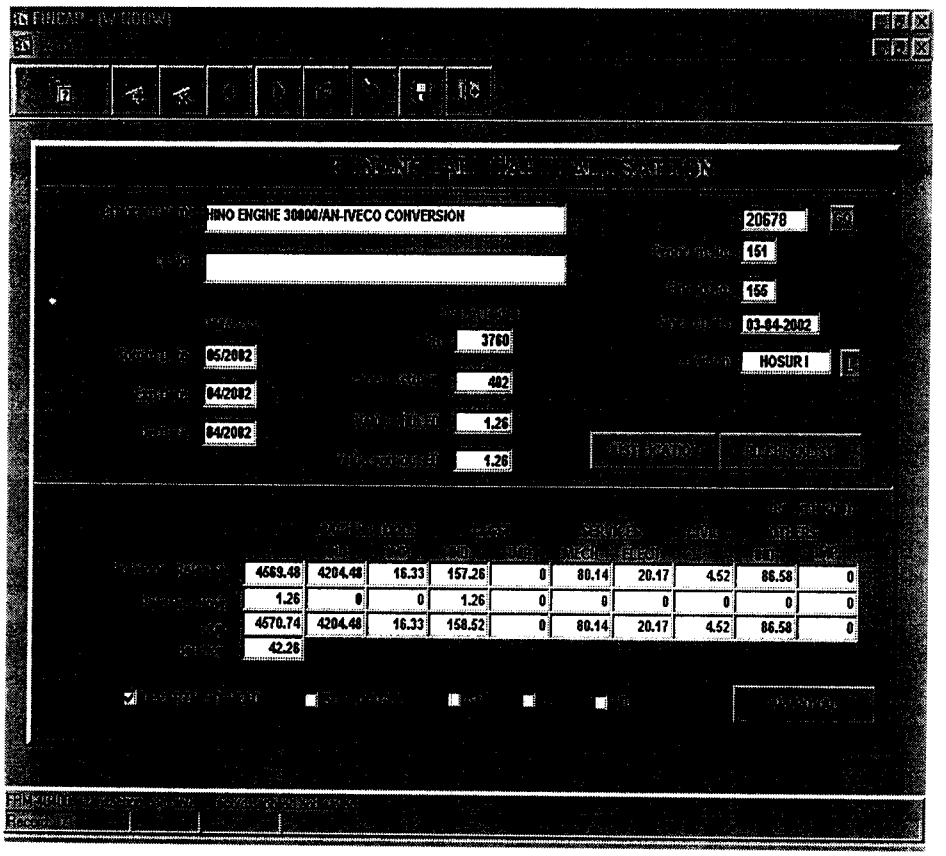

GB OPTIMISATION 37100
6
700000
4200000
TERMS AND CONDITIONS
ABOUT THE EXPORT BUSINESS IN SINGAPORE
ABOUT THE EXPORT
IMPORTER

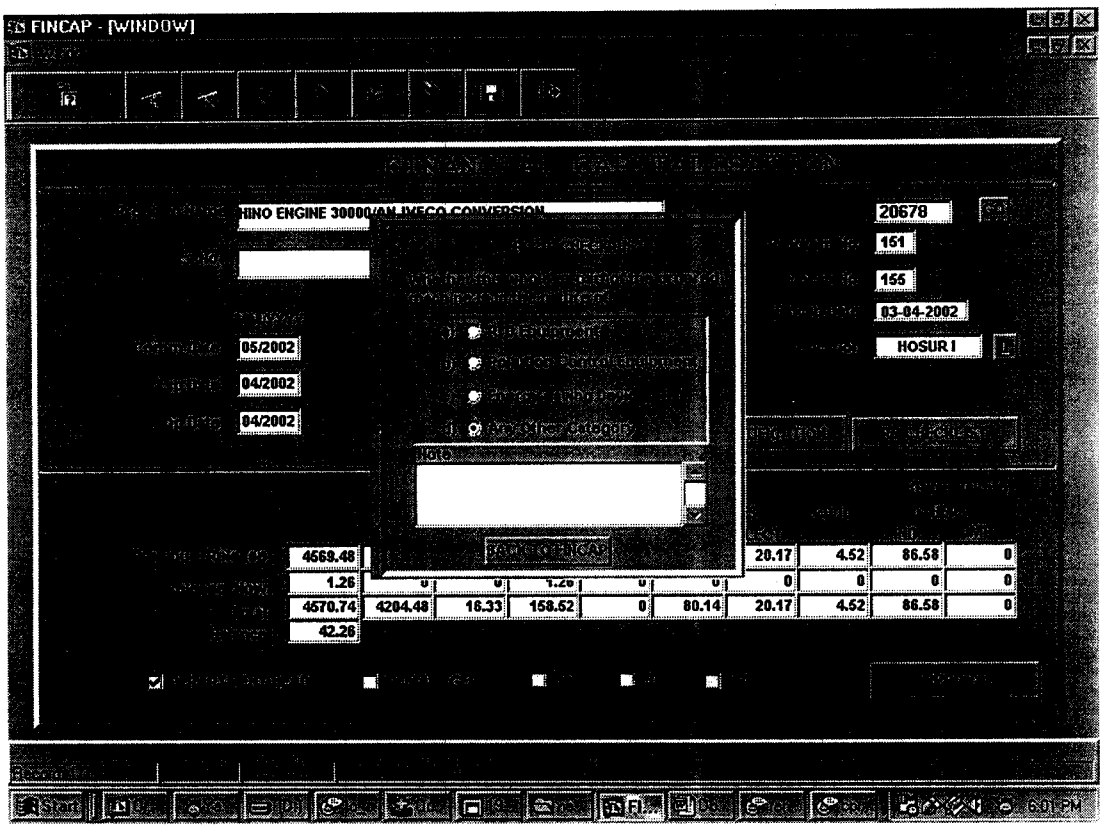

MODIFICATION
Old PO No
MODIFICATION
1209

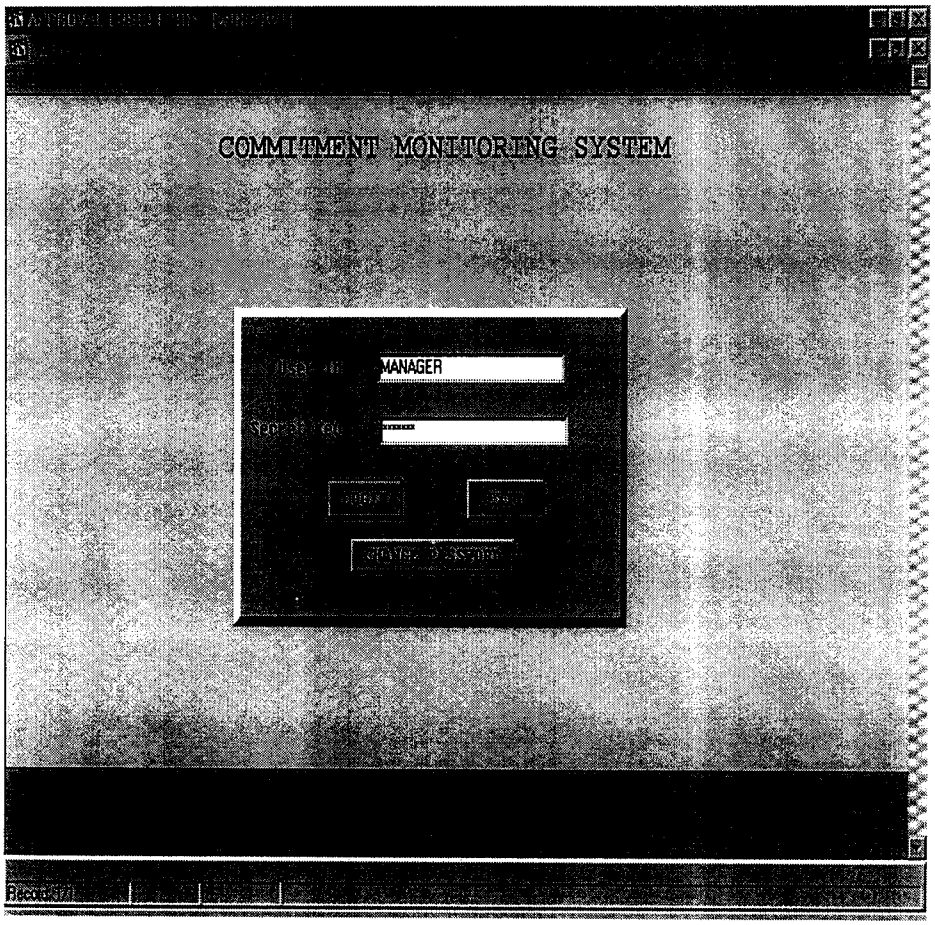

NOTE TO MD
149
07
02-04-2002
15000
B001
0
B001
0
0
0
0
5000
20,000.00
GO TO NEXT

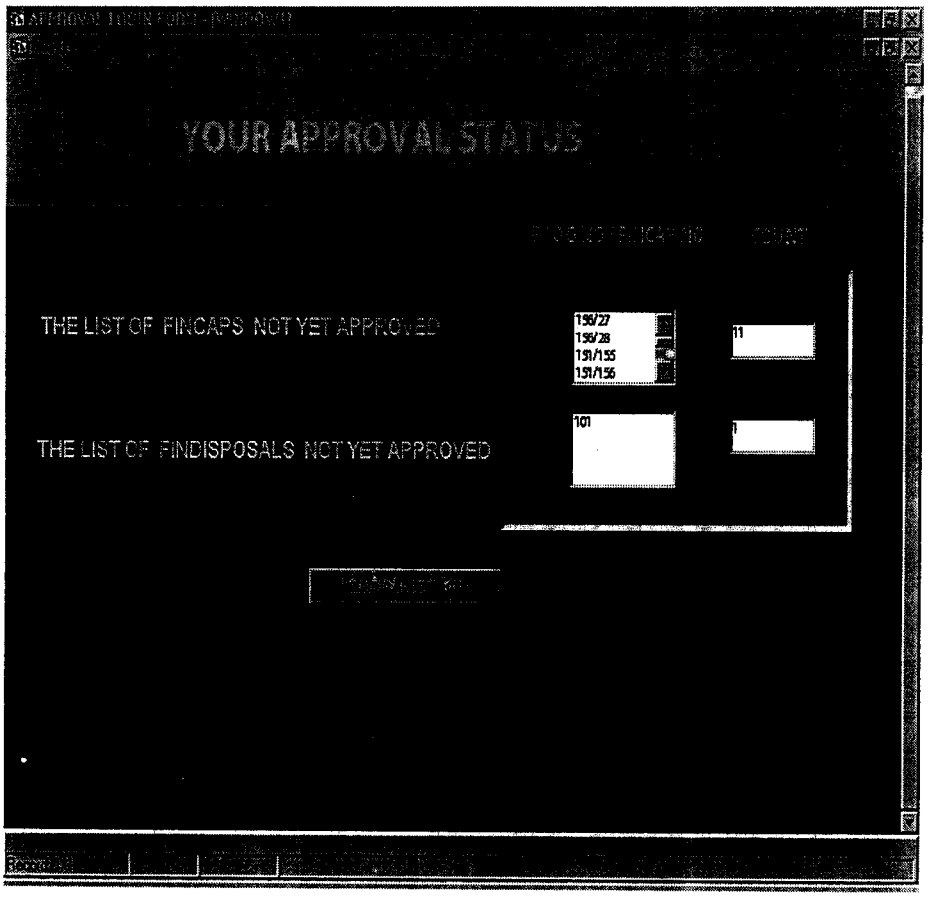

149
18-APR-2002
15,000.00
3,45,676.00
3,30,676.00
.00
.00
.00
.00
.00
.00
.00
.00
.00
.00
.00
.00
5,000.00
345.00
-4,655.00
20,000.00
3,46,021.00
3,26,021.00

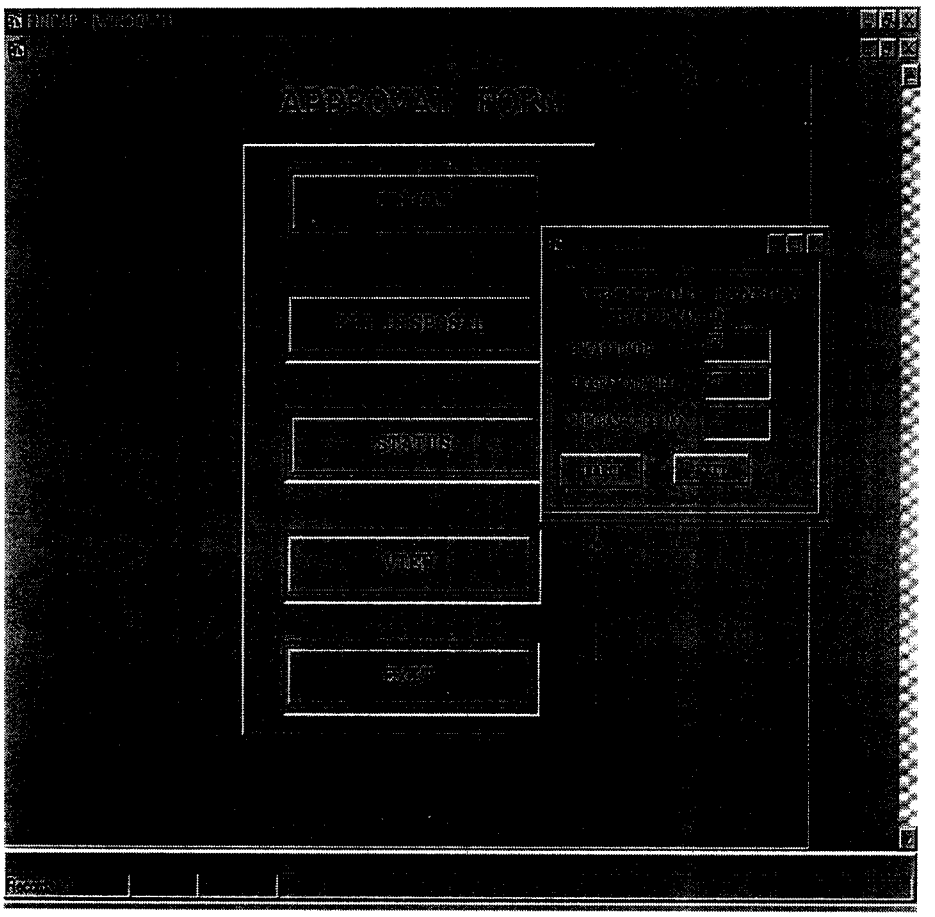

145
16-04-92
OLD SPARE PARTS OF INDOL MACHINE.
1990
15,000.00
15,000.00

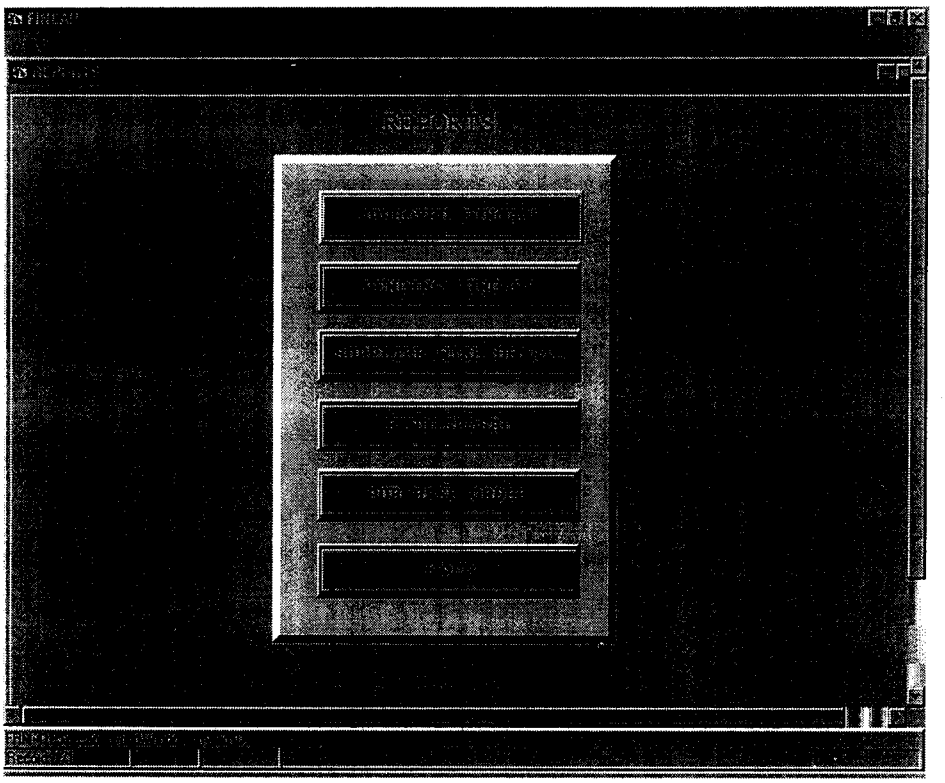

HINO ENGINE 30800/AN-IVECO CONVERSION
20678
151
155
03-04-2002
HOSUR I
3760
482
1.26
1.26
05/2002
04/2002
04/2002
4569.48 4204.48 16.33 157.26 0 80.14 20.17 4.52 86.58 0
1.26 0 0 1.26 0 0 0 0 0 0
4570.74 4204.48 16.33 158.52 0 80.14 20.17 4.52 86.58 0
42.26


FINCAP - [WINDOW]
HINO ENGINE 3000Q/AN IVECO CONVERSION
20678
151
155
03-04-2002
HOSUR I
05/2002
04/2002
04/2002
4569.48
1.26
4570.74
42.26
4204.48
16.33
158.52
0
80.14
20.17
4.52
86.58
0
Start


COMMITMENT MONITORING SYSTEM
MANAGER


YOUR APPROVAL STATUS
THE LIST OF FINCAPS NOT YET APPROVED
1:50/27
1:50/28
1:51/155
1:51/156
11
THE LIST OF FINDISPOSALS NOT YET APPROVED
101





REPORTS

SUPPLIERS DETAILS

Supp_code	Supplier Name	Prog no	Fincap no	Order no	Release now
I001	Inspiration	**151**	155	23220	1.26
S001	Southern Machine Tools		156	23231	.15
V001	Varalakshmi Agencies		157	23251	4.22
				23301	4.20
L001	Lokesh Machine Ltd		158	23302	.33
			159	23315	5.27
B001	BFW Ltd		160	25015	.93
S002	Southern Machine Tools		161	25016	.14
T001	Tawakkal Fabrn Works		162	25020	.2
U001	Unique Technologies	**156**	27	26080	.16
E001	Empire Machine Tools		28	26082	1.63
			Total		**8.49**