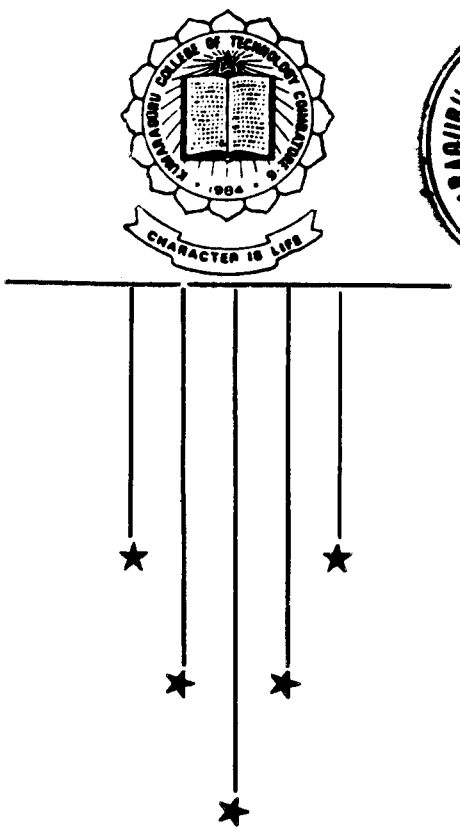
AUTOMATED ORDER PROCESSING FOR AN ENGINEERING INDUSTRY



PROJECT REPORT

Submitted by

M. C. GIREESH KUMAR
C. GOPALA KRISHNAN
N. RAMESH BABU
M. C. RAVI

Under the Guidance of

Ms. N. RAJATHI, B.E., M.I.S.T.E.,

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

1997 - 98

BACHELOR OF SCIENCE
(APPLIED SCIENCES-COMPUTER TECHNOLOGY)
OF THE BHARATHIAR UNIVERSITY, COIMBATORE

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Kumaraguru College of Technology

COIMBATORE-641 006.

KUMARAGURU COLLEGE OF TECHNOLOGY
COIMBATORE - 641 006

Department Of Computer Science And Engineering

CERTIFICATE

This Is To Certify That The Project Entitled

" Automated Order Processing For An Engineering Industry "

Has Been Submitted By

MR_____

In Partial Fulfillment Of The Requirements For The Award Of Degree Of Bachelor Of Science (Applied Sciences –Computer Technology) Branch Of The Bharathiar University, Coimbatore-641 046 During The Year 1997-98.

(Guide)

(Head Of The Department)

Certified That The Candidate Was Examined By Us In The Project Viva-Voce Examination Held

On _____ And The University Register Number Is _____.

(Internal Examiner)

(External Examiner)

28/4/98

CERTIFICATE

SPARTEK MARKETING AGENCY

11/16, SIVAJI COLONY EXTENSION,
EDAYARPALAYAM, COIMBATORE - 641 025
PH: 445544

To whom so ever it may concern:

This is to certify that the project work titled 'Automated order processing for an engineering industry' is the bona fide work done by the following students for our company

- Gireesh Kumar M.C
- Gopalakrishnan C.
- Ramesh Babu N.
- Ravi M.C

We commend them for their keen interest in the project, which helped us a lot in the automation of our organisation.

For SPARTEK MARKETING AGENCY

Proprietor 14/03/98

ACKNOWLEDGEMENT

We would like to thank our Principal **Dr. S.Subramanian, M.Sc (Engg)., Ph.D., S.M.I.E.E.E., M.I.S.T.E.,** and the college management, who extended us their fullest support.

We would also like to thank our Head of the Department, **Prof P. Shanmugam, B.E., M.Sc (Engg)., M.S. (Hawaii), S.M.I.E.E.E., M.I.S.T.E.,** for permitting us to do under take this project.

We are grateful to our guide, **Ms. N. Rajathi, B.E., M.I.S.T.E.,** for her most valuable and helpful guidance and patronage.

We acknowledge the encouragement of **Mr.Jacob, Spartek Marketing Agency,** who made it all possible, by letting us takeup this project.

We are thankful to all the faculty of our department, who offered us timely support and paved way for the completion of this project.

Lastly, we express our gratitude to our parents and friends who offered us their invaluable ideas and comments to better this project.

SYNOPSIS

SYNOPSIS

The Order processing system in an engineering industry employs a large number of operations that are done manually. The maintenance operations such as updation, computation and record keeping consume a lot of time and energy, thus leading to a large number of errors.

The problems encountered in a manual order processing system can be eliminated and the errors can be reduced to a large extent by employing the automated system. This system automatically maintains the records and processes the orders, thus reducing the scope for errors to creep in.

This project, being a GUI based application is user friendly, giving all possible convenience to the user. The user can perform all operations at the clicking of buttons.

The above system is implemented using MS-VISUAL BASIC 5.0 Enterprise as the front-end and MS-ACCESS 7.0 as the database engine.

CONTENTS

CONTENTS

INTRODUCTION

INTRODUCTION

ABOUT AUTOMATED ORDER PROCESSING

Manual order processing is a cumbersome process, which may result in inconsistency of data and also paves way for many errors to creep in. The whole idea of this project is to automate the process of order processing in order to make it user-friendly and make the process more efficient. This project reduces the time taken to process the orders and also greatly reduces the paperwork involved. The calculations are done automatically in order to reduce the errors that occur due to manual calculations.

The screens are designed to be user-friendly by making the controls self-explanatory. ToolTip texts are used to help the users understand what each control stands for. All the required operations involved in the order processing process can be done by simply clicking buttons on the screen and entering the required data. Messages are generated during each operation to help the user. The users no longer need to remember codes and numbers in order to enter data, all the possible codes and numbers relevant to a particular field is listed for the user to select the required code or number.

ORDER PROCESSING

ORDER PROCESSING

Orders are received by mail and phone at an engineering industry by the order-entry group. Both salesperson and customers initiate orders by telephone, telegram, or mail. Orders are sometimes delivered by customers and salesperson at the plant. Regardless of the source, the order-entry group prepares a sales order form from the information received. Each order is stamped with the time of receipt and is entered in the sales order log. The purpose of this log is to determine the elapsed time between order receipt and shipment. A similar shipment log is maintained in shipping so that elapsed time can be determined at any time on any order or combination of orders.

After preparation number of editing operations are performed on the sales order form. First, it is given credit check, utilizing the customer credit file. Second, the order is edited for clarification of the items groups ordered quantities, stock number, price, and customer account number. For this purpose the order is compared against the company catalog, price list, and customer list. A third check involves the billing and shipping addresses. These are verified by comparing them with the master customer file.

After editing, the sales order form is forwarded to the warehouse. Here the order is passed to the inventory clerk, who reduces the perpetual inventory record balance. If an item is out of stock, the ordered item is marked "PO" (for pending-order). The sales order then goes to Data Processing, where a copy of it is made. This copy is sent back to the warehouse, where it is filed in the warehouse sales order file pending return of the sales order from Data Processing.

In the order follow-up section the invoices are separated into two batches:

- Completed orders, which are filed in the closed-order file, and
- Incomplete orders, which are filed sequentially by order number in the open-order file.

Pending-ordered goods are sent to the warehouse, from Receiving if inside the company as a result of the back order purchase request or from Manufacturing if manufactured in the company's manufacturing facilities as a result of the back order production request. When merchandise arrives from either source, it is checked against purchase orders, purchase requests, and manufacturing requests, and the warehouse-receiving clerk notifies the order follow-up section. A follow-up clerk then searches the open-order file, and a new sales order is prepared for the pending-ordered items. This order is processed in the same way as a new completely filled order.

In addition to preparing a new order for the pending-ordered items, the old back order is placed in the closed-order file if the items completely fill the back order. If not, the open-order file copy is appropriately annotated with information concerning receipt of the partial order.

PURCHASE ORDERS

Material requisitions are received in purchase department by a clerk. After editing the requisitions for correctness and completeness, the clerk refers to a manual vendor's historical file that is maintained by vendor information. After the purchase manager selects a vendor he indicates the necessary information on the requisition, which is then passed to a typist for the creation of a purchase order. In general there is one purchase order prepared for each requisition. After the purchase manager edits and signs the purchase order, copies go to a clerk for mailing to the vendor, filling of a copy with the requisition in an open-order file for follow-up, a copy to the warehouse, and a copy to materials control in manufacturing.

Upon receipt of the requisition, the vendor/supplier dispatches the ordered goods to the company along with the invoice. The goods that are received are recorded by the parts-arrival record, and the inventory is duly updated. Periodic runs are made on the parts inventory to check the status of the inventory and orders are made as and when required.

MATERIAL STATUS SYSTEM

The two main purpose of this system are to generate an unordered requirements listing for input to the purchasing system and to provide a materials status and history report for use as follow-up and status information.

Each week, and more frequently in pressing situations, a computer parts requirement run is made to develop the new or changed parts requirements that have developed from the bill of materials system. This run uses the parts requirement file of the bill of materials system and material status file to create a purchase requirement list. The material-ordering section to prepare requisitions for purchasing then uses this list.

In addition to the requisitions for purchasing, material-ordering personnel also manually prepare a material order card that is subsequently keypunched for purposes of preparing material status cards. These are used as input to the daily computer material status run. This run merges the status card with the materials status file to produce the updated material status file and the materials status and history report. The file, and hence the report, is changed also from these events:

- Release of purchase order to vendor
- Receipt of materials in Receiving
- Material passed into the stockroom from inspection

SYSTEM REQUIREMENTS

Personal computer with a 486DX or higher processor running Microsoft Windows 95 or Windows NT 3.51 or later

- 8 megabytes (MB) of memory (16 MB recommended for Windows 95)
- 10 MB available hard disk space
- MPC2-compatible CD-ROM drive
- Super VGA or higher resolution video adapter capable of displaying 256 colors or more
- Microsoft Mouse or compatible pointing device

ABOUT VISUAL BASIC 5.0 ENTERPRISE:

The fastest and easiest way to create applications for Microsoft Windows:

So what is Visual Basic? The "Visual" part refers to the method used to create the graphical user interface (GUI). Rather than writing numerous lines of code to describe the appearance and location of interface elements, you simply drag and drop prebuilt objects into place on screen. If you've ever used a drawing program such as Paint, you already have most of the skills necessary to create an effective user interface.

The "Basic" part refers to the BASIC (Beginners All-Purpose Symbolic Instruction Code) language, a language used by more programmers than any other language in the history of computing. Visual Basic has evolved from the original BASIC language and now contains several hundred statements, functions, and keywords, many of which relate directly to the Windows GUI. Beginners can create useful applications by learning just a few of the keywords, yet the power of the language allows professionals to accomplish anything that can be accomplished using any other Windows programming language.

The Visual Basic programming language is not unique to Visual Basic. The Visual Basic programming system, Applications Edition included in Microsoft Excel, Microsoft Access, and many other Windows applications uses the same language. The Visual Basic programming system, Scripting Edition (VBScript) for Internet programming is a subset of the Visual Basic language. The investment you make in learning Visual Basic will carry over to these other areas.

Whether your goal is to create a small utility for yourself or your work group, a large enterprise-wide system, or even distributed applications spanning the globe via the Internet, Visual Basic has the tools you need.

- Data access features allow you to create databases and front-end applications for most database formats, including Microsoft SQL server and other enterprise-level databases.
- ActiveX™ Technologies allow you to use the functionality provided by other applications, such as Mircosoft Word wordprocessor, Microsoft Excel spreadsheet, and other Windows applications. You can even automate applications and objects created using the Professional or Enterprise editions of Visual Basic.
- Internet capabilities make it easy to provide access to documents and applications across the Internet from within your application.
- Your finished application is a true .exe file that uses a run-time dynamic-link library (DLL) that you can freely distribute.

The Guide to Data Access Objects provides detailed information about the Microsoft Jet database engine, the Data control, and the data access objects (DAO) programming interface.

Visual Basic lets you create and manipulate data access objects such as Database, Field, and Index objects, that correspond to the various parts of the physical database you want to access. You use the properties and methods of these objects to perform operations on the database.

DAO lets you write programs that access many different kinds of databases, including native databases that use the same format as Microsoft Access, external databases like FoxPro or Paradox, and ODBC client/server databases like Microsoft SQL Server.

Visual Basic Editions

Visual Basic is available in three versions, each geared to meet a specific set of development requirements.

1. The Visual Basic allows programmers to easily create powerful applications for Microsoft Windows 95 and Windows NT®. It includes all intrinsic controls, plus grid, tab, and data-bound controls.
2. The Professional edition provides computer professionals with a full-featured set of tools for developing solutions for others. It includes all the features of the Learning edition, plus additional ActiveX controls, including Internet controls, and the Crystal Report Writer. Documentation provided with the Professional edition includes the Programmer's Guide.
3. The Enterprise edition allows professionals to create robust distributed applications in a team setting. It includes all the features of the Professional edition, plus the Automation Manager, Component Manager, database management tools, the Microsoft Visual Source Safe™ projects-oriented version control system, and more.

ABOUT ACCESS 7.0

Access 7.0, generally called the "Jet Engine" in Visual Basic, is a versatile and powerful database manager (Backend) supplied with Visual Basic 5.0. Accees needs no ODBC to get connected with Visual Basic. Even though Access 7.0 provides many of the features provided by Visual Basic itself, it is found that the work the best together.

Access 7.0 provides user-friendly ways to design and modify databases. It hosts a wide range of data types as well. It is a RDBMS and has a built in SQL, as most RDBMS have.

Few of the terms used in Access 7.0 are:

Dynaset	: A table that is updatable.
Snapshot	: A table that cannot be updated.
Fields	: Columns of a table.
Records	: Row of a table.

Data from the databases can be accessed in Visual Basic through a special control called the 'Data Control'.

Some of the features available in Access 7.0 are given below:

- Standard database formats for many applications are in-built.
- Database creation is very simple and is done in a GUI environment.
- Modifications can be easily done to the created database.
- Inputs to the database can be formatted using the validation rule and errors can be generated for improper inputs using the validation text.

Summarizing the features of both Visual Basic and Access, we can see that they form a powerful and awesome tool to create complex applications for windows, together.

SYSTEM IMPLEMENTATION

SYSTEM IMPLEMENTATION

System implementation is the process where the system is actually given to the user and is tested in the environment it is going to be employed in. The system implementation involves the following stages:

- The system is tested with real time data and the results are verified for accuracy and efficiency. The developed application is tested for bugs and errors and are eliminated if found.
- The users are trained on the usage of the application.
- The system is modified based on user requirements to ensure user satisfaction.

Finally feedback is obtained from the users about the application and based on the user comments the application is changed to suite the users.

Our application has been tested with enough test data as well as real time data and has been found to work satisfactorily. The application has been thoroughly checked to determine any bugs and errors and most of the errors are cleared to ensure proper functioning of the application. Data entry validations are carried wherever required to ensure that correct data are entered into the system. The application is user-friendly and hence there is to a large extent, proper interaction with the user, and this ensures the entry of valid data only.

If there is any invalid data entry, the user is prompted to enter valid data into the system. Micro help is provided when the mouse pointer moves over any button to inform the user what operation can be done when the user presses that button. There is a host of other user-friendly features that makes the application easy to be operated. Even a layman can easily use the application.

CONCLUSION

CONCLUSION

This system has been designed and tested successfully with both test data and real data, and is satisfactorily implemented. This system is a GUI-based, user-interactive on-line system working under windows95. It provides facilities to maintain and manipulate data, retrieve information and to create reports quickly and effectively. The system makes the process of order processing very easy and efficient and also timesaving.

The automated system is implemented and is working well providing satisfaction to the users. The system is developed with high modularity and also a high level of compatibility is associated with it. This makes it easy to integrate the system with other systems such as Inventory Management or Production Management.

The system can be easily integrated with other compatible systems to ensure efficient and proper automation of the whole organization.

BIBLIOGRAPHY

BIBLIOGRAPHY

DBMS

1 Introduction to data base system- 3rd Edition ,vol 1
C.J.Date, Narosa Publications, 1994

2 Fundamentals of Data Base Systems- 2nd Edition
Elmasri & Navathe

MIS

1 Information Systems for Modern Management- 3rd Edition
Robert G. Murdick, Joel E. Ross & James R. Claggett

VISUAL BASIC

1 Mastering VISUAL BASIC 5.0- 1st Edition
Evangelos Petroutsos

2 Microsoft VISUAL BASIC 5.0 Enterprise Edition
Microsoft Manual

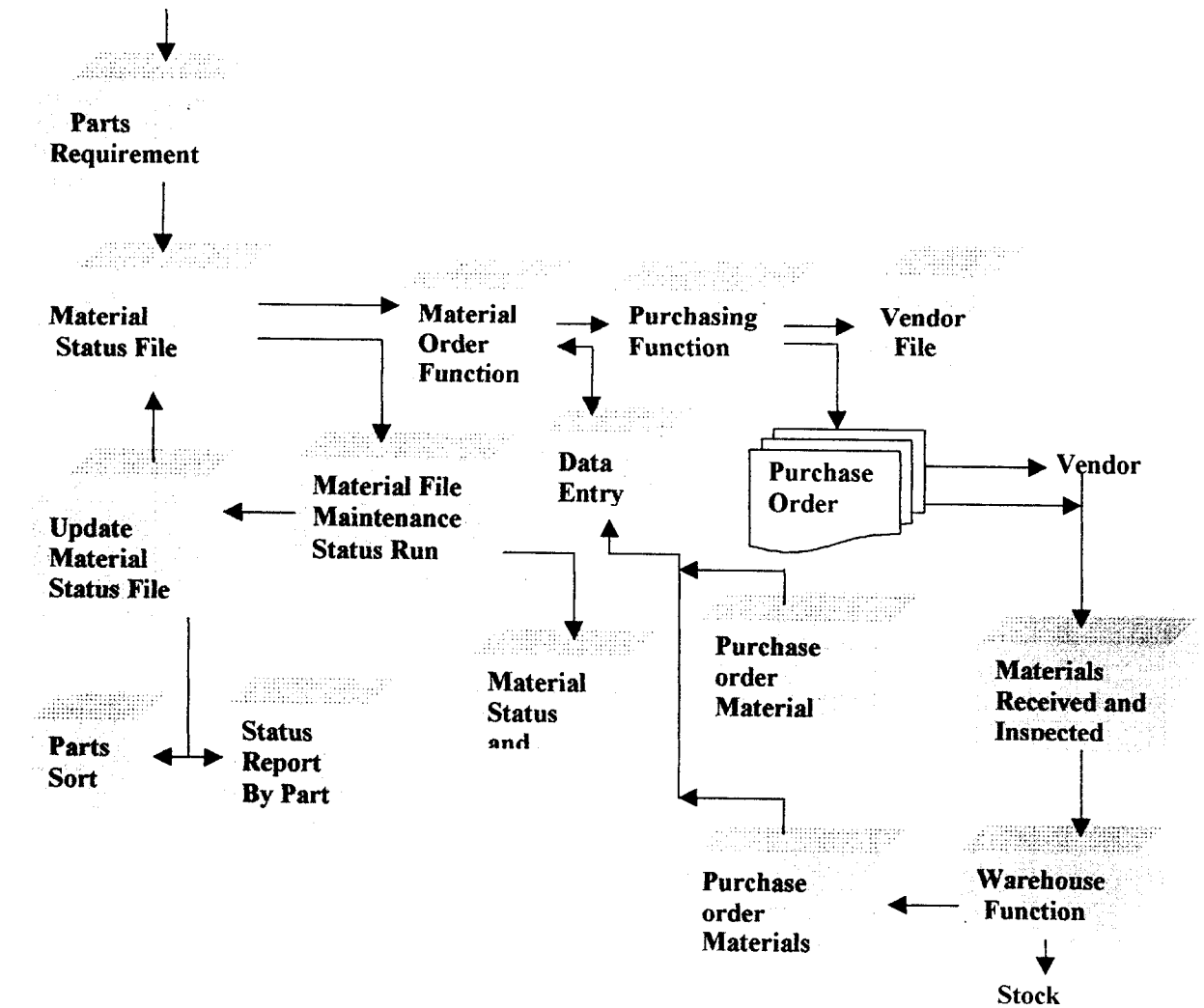
3 Devoloper's Guide VISUAL BASIC 5.0 1st Edition
Anthony T.Mann

APPENDIX

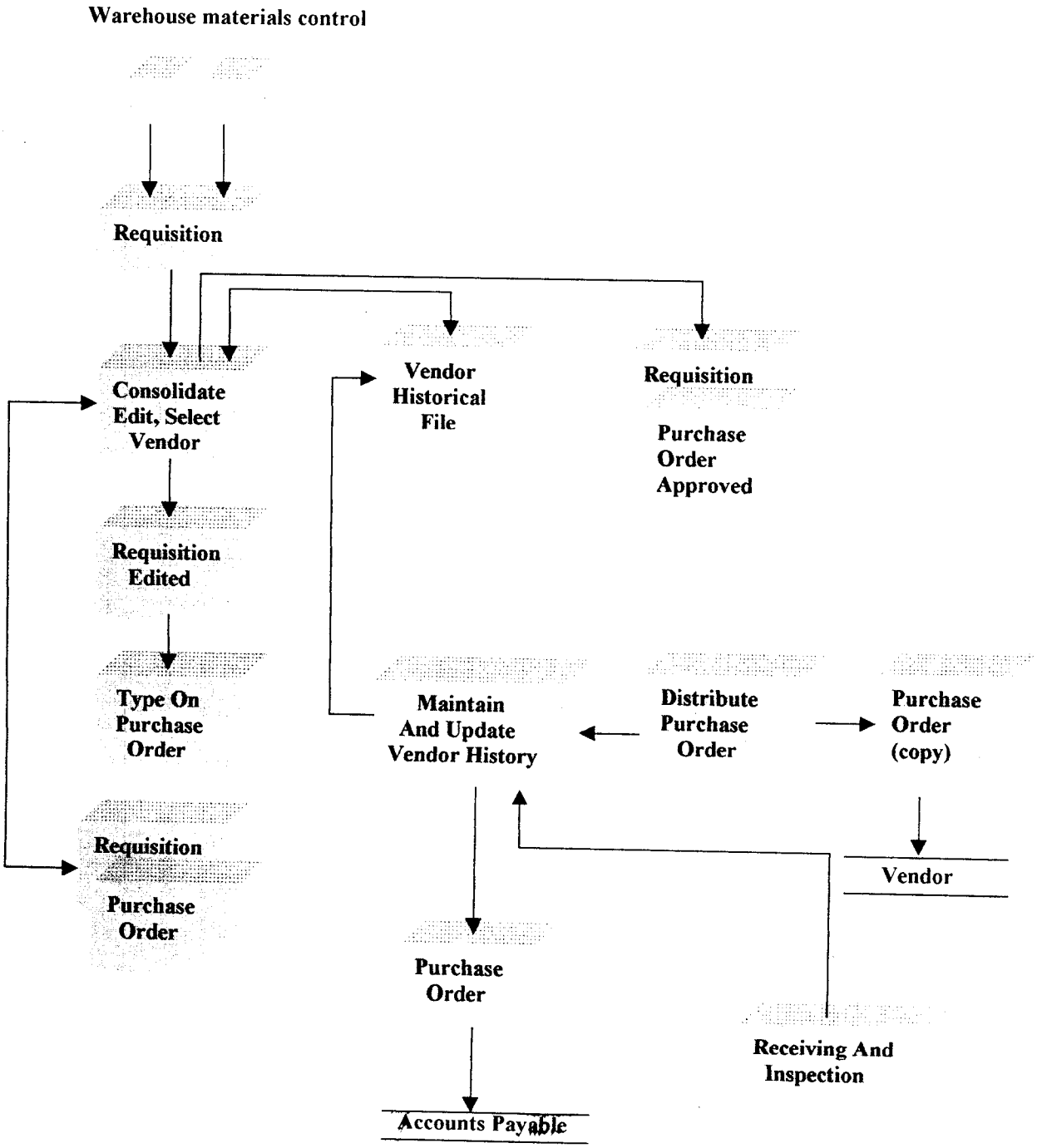
FLOWCHART

Material status system

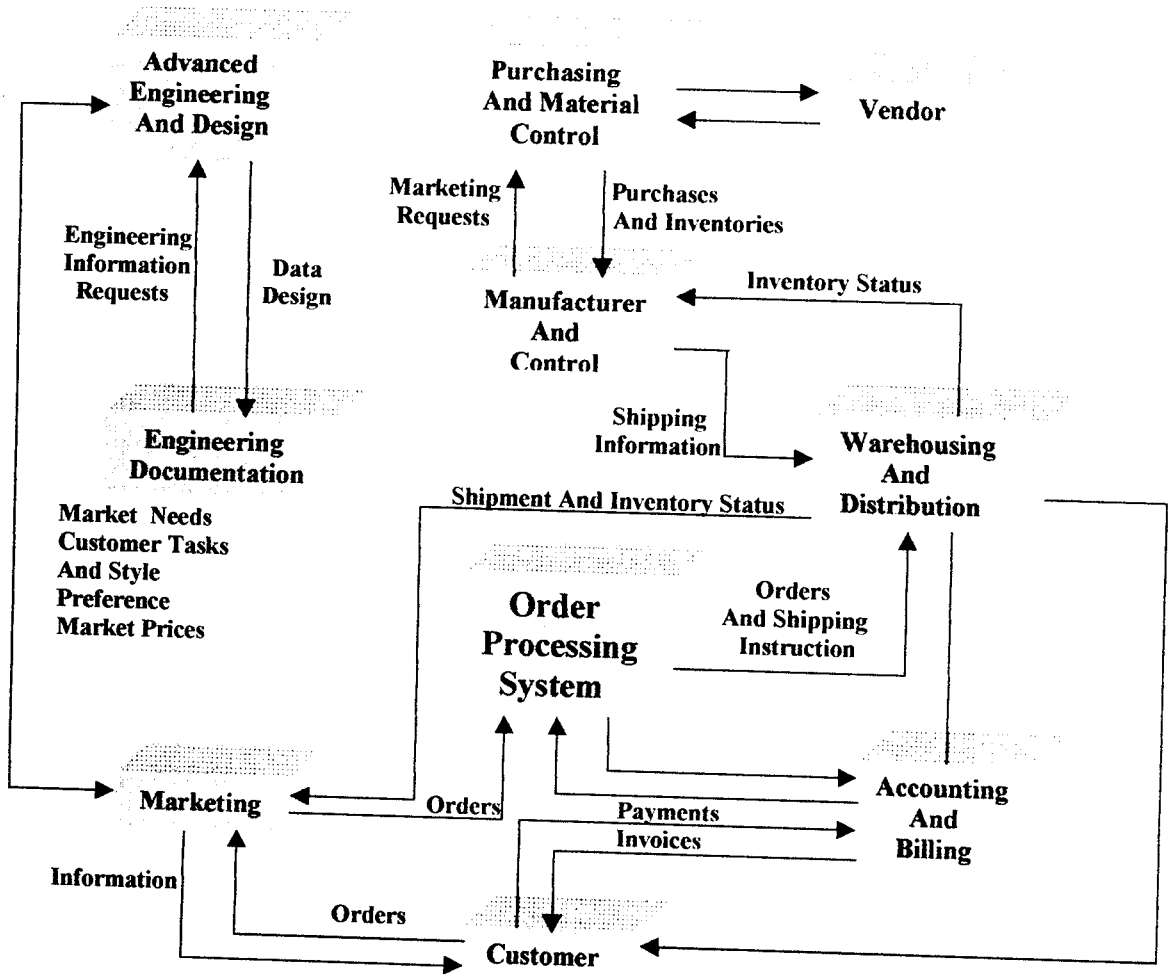
FORM ENGINEERING BILL
OF MATERIAL SYSTEM



Purchase system



Order Processing System



FORMS

FORMS

FRMMAIN

This is the main form of the project. It consists of menus to access the various other forms in the project.

FRMPASS

It is the login form where the user name and password should be entered in order to access the application.

FRMCPASS

This is the form used to change the password for the application.

FRMCOMPANY

This form is used to view and update the details of the company.

FRMCUSTOMER

This form is used to manipulate and view details of the customers.

FRMSUPPLIER

To manipulate and view details of the suppliers.

FRMPART_MASTER

To manipulate and view details of the parts.

FRMPRODUCT_MASTER

To manipulate and view details of the products.

FRMPARTS_ARRIVAL

To record the arrival of parts from the suppliers.

FRMSTORES_ARRIVAL

To record arrival of products to the warehouse from the production line.

FRMSALES_ENQUIRY

To register customer enquries into the system.

FRMSALES_ORDER

To register the orders received by the company.

FRMPURCHASE_ENQUIRY

To prepare enquiry for requesting quotation for parts.

FRMPURCHASE_ORDER

To prepare order for sending to the supplier.

FRMQUOTATION_DET

For preparing the quotation in response to the customers requests.

FRMMAT_RET_HEAD

To record the materials that are returned by the customers.

FRMPENDING_ORDER

To record the pending orders.

FRMINVOICE_DET

To prepare the invoice for the materials ordered by the customers.

TABLES

TABLES

QUOTATION DETAILS

QUOTATION_NO	TEXT	8	NOTNULL
PROD CODE	TEXT	8	NOTNULL
QUANTITY	LONG	4	
SNO	INTEGER	2	

QUOTATION HEADER

CUS_CODE	TEXT	8	NOTNULL
TAX	INTEGER	2	
PAYMENTTERMS	TEXT	20	
QUERY_NO	TEXT	8	NOTNULL
QUERY_DATE	DATE/TIME	8	
QUOTATION_NO	TEXT	8	NOTNULL
QUOTATION_DATE	DATE/TIME	8	
VALIDITY	INTEGER	2	

STORES ARRIVAL

SUP_CODE	TEXT	8	NOTNULL
DATE	DATE/TIME	8	
SAR_NO	TEXT	8	NOTNULL
PROD_CODE	TEXT	8	NOTNULL
QUANTITY	LONG	4	

SUPPLY MASTER

CODE	TEXT	8	NOTNULL
NAME	TEXT	30	
ADD1	TEXT	30	
ADD2	TEXT	30	
CITY	TEXT	30	
STATE	TEXT	30	
PIN	TEXT	6	
PHONES	TEXT	30	
FAX	TEXT	30	
TELEX	TEXT	20	
EMAIL	TEXT	20	
TNGST	TEXT	14	
CST	TEXT	14	
BANKER	TEXT	50	

SYSTEM

PASS	TEXT	15	

CUSTOMER

CODE	TEXT	8	NOTNULL
ADD1	TEXT	30	
ADD2	TEXT	30	
CITY	TEXT	30	
CREDIT_DAYS	CURRENCY	8	
CREDIT_LOW	CURRENCY	8	
CREDIT_UP	CURRENCY	8	
CST_NO	TEXT	14	
NAME	TEXT	30	
STATE	TEXT	30	
PIN	TEXT	6	
PHONES	TEXT	30	
FAX	TEXT	30	
TELEX	TEXT	20	

EMAIL	TEXT	20	
TYPE	TEXT	10	
BANKER	TEXT	50	
DEPOSIT	CURRENCY	8	
BALANCE	CURRENCY	8	
TNGST_NO	TEXT	14	

DESTINATION MASTER

ADD1	TEXT	30	
ADD2	TEXT	30	
CITY	TEXT	30	
CODE	TEXT	3	NOTNULL
CONTACT_DESIGN	TEXT	30	
CONTACT_PERSON	TEXT	30	
CUS_CODE	TEXT	8	NOTNULL
EMAIL	TEXT	20	
FAX	TEXT	30	
PHONE	TEXT	30	
PIN	TEXT	6	
TELEX	TEXT	20	

INVOICE MASTER

FREIGHT	TEXT	20	
INVOICE_DATEDATE/TIME	8		
INVOICE_NO	TEXT	8	NOTNULL
NO_OF_CASESLONG4			
ORDER_NO	TEXT	8	NOTNULL

MATERIAL RETURN DETAILS

PRICE	CURRENCY	8	
PROD_CODE	TEXT	8	NOTNULL
QUANTIY	LONG	4	
RET_NO	TEXT	5	NOTNULL
SNO	INTEGER	2	

MATERIAL RETURN HEADER

DATE	DATE/TIME	8	
INVOICE_NO	TEXT	8	NOTNULL
ORDER_NO	TEXT	8	NOTNULL
REASON	TEXT	50	
RET_NO	TEXT	5	NOTNULL
VALUE	CURRENCY	8	

ORDER DETAILS

ORDER_NO	TEXT	8	NOTNULL
PRICECURRENCY8			
PROD_CODE	TEXT	8	NOTNULL
QUANTITYLONG4			
SNOINTEGER2			

ORDER HEADER

CUS_CODE	TEXT	8	NOTNULL
CUS_ORD_NO	TEXT	8	
DATE	DATE/TIME	8	
DESPATCH_DEST_CODE	TEXT	5	NOTNULL
DISCOUNTINTEGER2			
ORDER_NO	TEXT	8	NOTNULL
SPARE_VALUE	CURRENCY	8	
SPECIAL_INST	TEXT	50	
TAX	INTEGER	2	

OUR DETAILS

ADD1	TEXT	30	
ADD2	TEXT	30	
CITY	TEXT	30	
CST	TEXT	14	
EMAIL	TEXT	20	
FAX	TEXT	30	

NAME	TEXT	30	
PHONES	TEXT	30	
PIN	TEXT	6	
STATE	TEXT	30	
TELEX	TEXT	20	
TNGST	TEXT	14	

PART MASTER

DESCRIPT	TEXT	50	
E_O_Q	TEXT	4	
PART_CODE	TEXT	8	NOTNULL
STOCK_LEVELLONG4			
REORDER_LEVELLONG4			
REORDER_STATUS	TEXT	3	
U_O_M	TEXT	15	

PARTS ARRIVAL

DATE	DATE/TIME	8	
PARNO	TEXT	8	NOTNULL
PART_CODE	TEXT	8	NOTNULL
QUANTITY	LONG	4	
SUP_CODE	TEXT	8	NOTNULL

PENDING ORDER DETAILS

BALANCE	LONG	4	
PENDING_NO	TEXT	5	NOTNULL
PROD_CODE	TEXT	8	NOTNULL
QUANTITY	LONG	4	
SNO	INTEGER	2	
SUPPLIED	LONG	4	

PENDING ORDER HEADER

DAYS_PENDING	INTEGER	2	
ORDER_NO	TEXT	8	NOTNULL
PENDING_NO	TEXT	5	NOTNULL

PRODUCT MASTER

DESCRIPT	TEXT	50	
PROD_CODE	TEXT	8	NOTNULL
RATE	CURRENCY	8	
STOCK_LEVEL	LONG	4	
U_O_M	TEXT	20	

PURCHASE EQUIRY DETAILS

NUM	TEXT	8	NOTNULL
SNO	INTEGER	2	
SPARE_CODE	TEXT	8	NOTNULL
QUANTITYLONG	4		

PURCHASE ENQUIRY HEADER

DATE	DATE/TIME	8	
NUM	TEXT	8	NOTNULL
SUP_CODE	TEXT	8	NOTNULL

PURCHASE ORDER DETAILS

NUM	TEXT	8	NOTNULL
SNO	INTEGER	2	
SPARE_CODE	TEXT	8	NOTNULL
QUANTITY	LONG	4	
RATE	CURRENCY	8	

PURCHASE ORDER HEADER

DATE	DATE/TIME	8	
DISCOUNT	INTEGER	2	
INSTRUCT	TEXT	50	
NUM	TEXT	8	NOTNULL
QUOT_NO	TEXT	8	NOTNULL
SUP_CODE	TEXT	8	NOTNULL
TAX	INTEGER	2	
VALUE	CURRENCY	8	

QUERY DETAILS

QUANTITY	LONG	4	
PROD_CODE	TEXT	8	NOTNULL
QUERY_NO	TEXT	8	NOTNULL
SNO	INTEGER	2	

QUERY HEADER

CUS_CODE	TEXT	8	NOTNULL
QUERY_DATE	DATE/TIME	8	
QUERY_NO	TEXT	10	NOTNULL

SAMPLE CODE

SAMPLE CODE FOR FORM

```
VERSION 5.00
Begin VB.Form FRMCUSTOMER
   BackColor          =  &H00C0C0C0&
   BorderStyle        =  1  'Fixed Single
   Caption            =  "Customer Details"
   ClientHeight       =  6795
   ClientLeft         =  -375
   ClientTop          =  -1530
   ClientWidth        =  9480
   FillColor          =  &H00FFFFFF&
   ForeColor          =  &H00000000&
   LinkTopic          =  "Form2"
   MaxButton          =  0  'False
   MinButton          =  0  'False
   ScaleHeight        =  6795
   ScaleWidth         =  9480
   StartUpPosition    =  1  'CenterOwner
   WindowState        =  2  'Maximized
   Begin VB.Frame Frame1
      Caption         =  "Type"
BeginProperty Font

         Name         =  "Times New Roman"
         Size         =  9
         Charset      =  0
         Weight       =  400
         Underline    =  0  'False
         Italic       =  0  'False
         Strikethrough =  0  'False
      EndProperty
      Height          =  375
      Left            =  5160
      TabIndex        =  11
      ToolTipText     =  "Customer Type Is Permanent"
      Top             =  720
      Width           =  4095
```

```
Begin VB.OptionButton Option2
   BackColor       =  &H00C0C0C0&
   Caption         =  "Permanent"
   BeginProperty Font
      Name         =  "Times New Roman"
      Size         =  9
      Charset      =  0
      Weight       =  400
      Underline    =  0   'False
      Italic       =  0   'False
      Strikethrough =  0   'False
   EndProperty
   Height          =  195
   Left            =  2400
   TabIndex        =  13
   Top             =  120
   Width           =  1335
End
Begin VB.OptionButton Option1
   BackColor       =  &H00C0C0C0&
   Caption         =  "Temporary"
   BeginProperty Font
      Name         =  "Times New Roman"
      Size         =  9
      Charset      =  0
      Weight       =  400
      Underline    =  0   'False
      Italic       =  0   'False
      Strikethrough =  0   'False
   EndProperty
   Height          =  195
   Left            =  720
   TabIndex        =  12
   ToolTipText     =  "Customer Type Is Temporary"
   Top             =  120
   Width           =  1335
End
End
Begin VB.CommandButton CMDCONFIRM
   Caption         =  "&Confirm"
   Default         =  -1  'True
   Enabled         =  0   'False
   BeginProperty Font
      Name         =  "Times New Roman"
```

```
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  Height            = 405
  Left              = 5880
  TabIndex          = 23
  ToolTipText       = "Press To Confirm Action"
  Top               = 6120
  Width             = 975
End
Begin VB.CommandButton CMDUPDATE
  Caption           = "&Modify"
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  Height            = 405
  Left              = 4800
  TabIndex          = 31
  ToolTipText       = "Modify Current Record"
  Top               = 6120
  Width             = 975
End
Begin VB.CommandButton CMDREMOVE
  Caption           = "&Remove"
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  Height            = 405
```

```
  Left            = 3720
  TabIndex        = 30
  ToolTipText     = "Remove Current Record"
  Top             = 6120
  Width           = 975
End
Begin VB.CommandButton CMDADD
  Caption         = "&Add"
  BeginProperty Font
    Name          = "Times New Roman"
    Size          = 9.75
    Charset       = 0
    Weight        = 400
    Underline     = 0  'False
    Italic        = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Height          = 405
  Left            = 2640
  TabIndex        = 29
  ToolTipText     = "Add NEW Record"
  Top             = 6120
  Width           = 975
End
Begin VB.CommandButton CMDMAIN
  Cancel          = -1  'True
  Caption         = "Ma&in"
  BeginProperty Font
    Name          = "Times New Roman"
    Size          = 9.75
    Charset       = 0
    Weight        = 400
    Underline     = 0  'False
    Italic        = 0  'False
    Strikethrough = 0  'False
  EndProperty
  Height          = 405
  Left            = 6960
  TabIndex        = 28
  ToolTipText     = "Go To Main"
  Top             = 5640
  Width           = 975
End
Begin VB.CommandButton CMDSEARCH
```

```
  Caption         =  "&Search"
  BeginProperty Font
    Name          =  "Times New Roman"
    Size          =  9.75
    Charset       =  0
    Weight        =  400
    Underline     =  0  'False
    Italic        =  0  'False
    Strikethrough =  0  'False
  EndProperty
  Height          =  405
  Left            =  5880
  TabIndex        =  27
  ToolTipText     =  "Find Record"
  Top             =  5640
  Width           =  975
End
Begin VB.CommandButton CMDLAST
  Caption         =  "&Last"
  BeginProperty Font
    Name          =  "Times New Roman"
    Size          =  9.75
    Charset       =  0
    Weight        =  400
    Underline     =  0  'False
    Italic        =  0  'False
    Strikethrough =  0  'False
  EndProperty
  Height          =  405
  Left            =  4800
  TabIndex        =  26
  ToolTipText     =  "Move To Last Record"
  Top             =  5640
  Width           =  975
End
Begin VB.CommandButton CMDNEXT
  Caption         =  "&Next"
  BeginProperty Font
    Name          =  "Times New Roman"
    Size          =  9.75
    Charset       =  0
    Weight        =  400
    Underline     =  0  'False
    Italic        =  0  'False
```

```
    Strikethrough   = 0  'False
  EndProperty
  Height            = 405
  Left              = 3720
  TabIndex          = 25
  ToolTipText       = "Move To Next Record"
  Top               = 5640
  Width             = 975
End
Begin VB.CommandButton CMDPREVIOUS
  Caption           = "&Previous"
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  Height            = 405
  Left              = 2640
  TabIndex          = 24
  ToolTipText       = "Move To Previous Record"
  Top               = 5640
  Width             = 975
End
Begin VB.CommandButton CMDFIRST
  Caption           = "&First"
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  Height            = 405
  Left              = 1560
  TabIndex          = 22
  ToolTipText       = "Move To First Record"
  Top               = 5640
  Width             = 975
```

```
End
Begin VB.Data data1
  Caption         =  " "
  Connect         =  "Access"
  DatabaseName    =  "D:\RAVI\spare.mdb"
  DefaultCursorType=  0  'DefaultCursor
  DefaultType     =  2  'UseODBC
  Exclusive       =  0   'False
  Height          =  345
  Left            =  0
  Options         =  0
  ReadOnly        =  0   'False
  RecordsetType   =  1  'Dynaset
  RecordSource    =  "select * from [CUSTOMER] Order by [CODE]"
  Top             =  6450
  Visible         =  0   'False
  Width           =  1200
End
Begin VB.TextBox txtFields
  BackColor       =  &H00FFFFFF&
  DataField       =  "CST_NO"
  DataSource      =  "data1"
  Height          =  350
  Index           =  19
  Left            =  5880
  MaxLength       =  14
  TabIndex        =  21
  Top             =  4560
  Width           =  3375
End
Begin VB.TextBox txtFields
  BackColor       =  &H00FFFFFF&
  DataField       =  "TNGST_NO"
  DataSource      =  "data1"
  Height          =  350
  Index           =  18
  Left            =  5880
  MaxLength       =  14
  TabIndex        =  20
  Top             =  4080
  Width           =  3375
End
Begin VB.TextBox txtFields
  BackColor       =  &H00FFFFFF&
```

```
  DataField       = "BALANCE"
  DataSource      = "data1"
  Height          = 350
  Index           = 17
  Left            = 5880
  TabIndex        = 19
  Top             = 3600
  Width           = 1935
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "CREDIT_DAYS"
  DataSource      = "data1"
  Height          = 350
  Index           = 16
  Left            = 5880
  TabIndex        = 18
  Top             = 3120
  Width           = 1935
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "DEPOSIT"
  DataSource      = "data1"
  Height          = 350
  Index           = 15
  Left            = 5880
  TabIndex        = 17
  Top             = 2640
  Width           = 1935
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "CREDIT_UP"
  DataSource      = "data1"
  Height          = 350
  Index           = 14
  Left            = 5880
  TabIndex        = 16
  Top             = 2160
  Width           = 1935
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
```

```
  DataField       = "CREDIT_LOW"
  DataSource      = "data1"
  Height          = 350
  Index           = 13
  Left            = 5880
  TabIndex        = 15
  Top             = 1680
  Width           = 1935
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "BANKER"
  DataSource      = "data1"
  Height          = 350
  Index           = 12
  Left            = 5880
  MaxLength       = 50
  TabIndex        = 14
  Top             = 1200
  Width           = 3375
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "EMAIL"
  DataSource      = "data1"
  Height          = 350
  Index           = 10
  Left            = 960
  MaxLength       = 20
  TabIndex        = 9
  Top             = 4560
  Width           = 3375
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "TELEX"
  DataSource      = "data1"
  Height          = 350
  Index           = 9
  Left            = 960
  MaxLength       = 20
  TabIndex        = 8
  Top             = 4080
  Width           = 3375
```

```
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "FAX"
  DataSource      = "data1"
  Height          = 350
  Index           = 8
  Left            = 960
  MaxLength       = 30
  TabIndex        = 7
  Top             = 3600
  Width           = 3375
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "PHONE"
  DataSource      = "data1"
  Height          = 350
  Index           = 7
  Left            = 960
  MaxLength       = 30
  TabIndex        = 6
  Top             = 3120
  Width           = 3375
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "PIN"
  DataSource      = "data1"
  Height          = 350
  Index           = 6
  Left            = 960
  MaxLength       = 6
  TabIndex        = 5
  Top             = 2640
  Width           = 3375
End
Begin VB.TextBox txtFields
  BackColor       = &H00FFFFFF&
  DataField       = "STATE"
  DataSource      = "data1"
  Height          = 350
  Index           = 5
  Left            = 960
```

```
  MaxLength     = 20
  TabIndex      = 4
  Top           = 2160
  Width         = 3375
End
Begin VB.TextBox txtFields
  BackColor     = &H00FFFFFF&
  DataField     = "CITY"
  DataSource    = "data1"
  Height        = 350
  Index         = 4
  Left          = 960
  MaxLength     = 30
  TabIndex      = 3
  Top           = 1680
  Width         = 3375
End
Begin VB.TextBox txtFields
  BackColor     = &H00FFFFFF&
  DataField     = "ADD2"
  DataSource    = "data1"
  Height        = 350
  Index         = 3
  Left          = 960
  MaxLength     = 30
  TabIndex      = 2
  Top           = 1200
  Width         = 3375
End
Begin VB.TextBox txtFields
  BackColor     = &H00FFFFFF&
  DataField     = "ADD1"
  DataSource    = "data1"
  Height        = 350
  Index         = 2
  Left          = 960
  MaxLength     = 30
  TabIndex      = 1
  Top           = 720
  Width         = 3375
End
Begin VB.TextBox txtFields
  BackColor     = &H00FFFFFF&
  DataField     = "NAME"
```

```
  DataSource     = "data1"
  Height         = 350
  Index          = 1
  Left           = 960
  MaxLength      = 30
  TabIndex       = 0
  Top            = 255
  Width          = 3375
End
Begin VB.TextBox txtFields
  BackColor      = &H00FFFFFF&
  DataField      = "CODE"
  DataSource     = "data1"
  Height         = 350
  Index          = 0
  Left           = 5880
  MaxLength      = 8
  TabIndex       = 10
  Top            = 240
  Width          = 3375
End
Begin VB.Shape Shape2
  BorderWidth    = 2
  Height         = 1335
  Left           = 120
  Top            = 5400
  Width          = 9375
End
Begin VB.Shape Shape1
  BorderWidth    = 2
  Height         = 5175
  Left           = 120
  Top            = 120
  Width          = 9375
End
Begin VB.Label lblLabels
  Alignment      = 1 'Right Justify
  BackColor      = &H00C0C0C0&
  BackStyle      = 0 'Transparent
  Caption        = "CST "
  BeginProperty Font
    Name         = "Times New Roman"
    Size         = 9.75
    Charset      = 0
```

```
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 19
  Left              = 4560
  TabIndex          = 50
  Top               = 4680
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "TNGST "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 18
  Left              = 4560
  TabIndex          = 49
  Top               = 4200
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Balance "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
```

```
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 17
  Left              = 4560
  TabIndex          = 48
  Top               = 3720
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Credit Days "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 16
  Left              = 4440
  TabIndex          = 47
  Top               = 3240
  Width             = 1335
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Deposit "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
```

```
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 15
  Left              = 4560
  TabIndex          = 46
  Top               = 2760
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Credit Up"
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 14
  Left              = 4560
  TabIndex          = 45
  Top               = 2280
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Credit Low "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
```

```
    Weight          =  400
    Underline       =  0  'False
    Italic          =  0  'False
    Strikethrough   =  0  'False
  EndProperty
  ForeColor         =  &H00000000&
  Height            =  255
  Index             =  13
  Left              =  4560
  TabIndex          =  44
  Top               =  1800
  Width             =  1215
End
Begin VB.Label lblLabels
  Alignment         =  1  'Right Justify
  BackColor         =  &H00C0C0C0&
  BackStyle         =  0  'Transparent
  Caption           =  "Banker  "
  BeginProperty Font
    Name            =  "Times New Roman"
    Size            =  9.75
    Charset         =  0
    Weight          =  400
    Underline       =  0  'False
    Italic          =  0  'False
    Strikethrough   =  0  'False
  EndProperty
  ForeColor         =  &H00000000&
  Height            =  255
  Index             =  12
  Left              =  4560
  TabIndex          =  43
  Top               =  1320
  Width             =  1215
End
Begin VB.Label lblLabels
  Alignment         =  1  'Right Justify
  BackColor         =  &H00C0C0C0&
  BackStyle         =  0  'Transparent
  Caption           =  "Email "
  BeginProperty Font
    Name            =  "Times New Roman"
    Size            =  9.75
    Charset         =  0
```

```
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 10
  Left              = -360
  TabIndex          = 42
  Top               = 4680
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Telex "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 9
  Left              = -360
  TabIndex          = 41
  Top               = 4200
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Fax "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
```

```
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 8
  Left              = -360
  TabIndex          = 40
  Top               = 3720
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Phone "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 7
  Left              = -360
  TabIndex          = 39
  Top               = 3240
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Pin "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
```

```
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 6
  Left              = -360
  TabIndex          = 38
  Top               = 2760
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "State "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 5
  Left              = -360
  TabIndex          = 37
  Top               = 2280
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "City "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
```

```
    Weight          =  400
    Underline       =  0  'False
    Italic          =  0  'False
    Strikethrough   =  0  'False
  EndProperty
  ForeColor         =  &H00000000&
  Height            =  255
  Index             =  4
  Left              =  -360
  TabIndex          =  36
  Top               =  1800
  Width             =  1215
End
Begin VB.Label lblLabels
  Alignment         =  1  'Right Justify
  BackColor         =  &H00C0C0C0&
  BackStyle         =  0  'Transparent
  Caption           =  "Add2 "
  BeginProperty Font
    Name            =  "Times New Roman"
    Size            =  9.75
    Charset         =  0
    Weight          =  400
    Underline       =  0  'False
    Italic          =  0  'False
    Strikethrough   =  0  'False
  EndProperty
  ForeColor         =  &H00000000&
  Height            =  255
  Index             =  3
  Left              =  -360
  TabIndex          =  35
  Top               =  1320
  Width             =  1215
End
Begin VB.Label lblLabels
  Alignment         =  1  'Right Justify
  BackColor         =  &H00C0C0C0&
  BackStyle         =  0  'Transparent
  Caption           =  "Add1 "
  BeginProperty Font
    Name            =  "Times New Roman"
    Size            =  9.75
    Charset         =  0
```

```
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 2
  Left              = -360
  TabIndex          = 34
  Top               = 840
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Name "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
    Weight          = 400
    Underline       = 0  'False
    Italic          = 0  'False
    Strikethrough   = 0  'False
  EndProperty
  ForeColor         = &H00000000&
  Height            = 255
  Index             = 1
  Left              = -360
  TabIndex          = 33
  Top               = 360
  Width             = 1215
End
Begin VB.Label lblLabels
  Alignment         = 1  'Right Justify
  BackColor         = &H00C0C0C0&
  BackStyle         = 0  'Transparent
  Caption           = "Code "
  BeginProperty Font
    Name            = "Times New Roman"
    Size            = 9.75
    Charset         = 0
```

```
      Weight          =  400
      Underline       =  0  'False
      Italic          =  0  'False
      Strikethrough   =  0  'False
    EndProperty
    ForeColor         =  &H00000000&
    Height            =  255
    Index             =  0
    Left              =  4440
    TabIndex          =  32
    Top               =  360
    Width             =  1215
  End
End
Attribute VB_Name = "FRMCUSTOMER"
Attribute VB_GlobalNameSpace = False
Attribute VB_Creatable = False
Attribute VB_PredeclaredId = True
Attribute VB_Exposed = False
```

SAMPLE SOURCE CODE

CODE FOR FRMCUSTOMER

```
Dim STATUS As Boolean
Private Sub CMDADD_Click()
Dim I As Integer, VAL As Integer, X As Integer
Dim STR As String
txtFields(1).SetFocus
Call ADDSTAT(Me)
data1.Recordset.AddNew
STATUS = True
I = MsgBox(" Press 'CONFIRM' After Entering Data ", vbInformation)
End Sub


Private Sub CMDCONFIRM_Click()
Dim VAL As Integer, I As Integer, X As Integer
Dim STR As String
VAL = MsgBox(" Are You Sure You Want To Update This Record ", vbYesNo +
vbQuestion)
If VAL = vbYes Then
LCODE: If txtFields(0).Text = "" Then
STR = InputBox(" 'CODE' Should Contain A Value , Enter A Value", vbCritical)
txtFields(0).Text = STR
GoTo LCODE
End If
If Option2.Value Then
data1.Recordset(11) = "PERMANENT"
For I = 13 To 17
LINE1: If Not IsNumeric(txtFields(I).Text) Then
STR = InputBox(lblLabels(I).Caption + " Should Contain A Numeric Value , Please Enter
One ")
txtFields(I).Text = STR
GoTo LINE1
End If
Next
Else: data1.Recordset(11) = "TEMPORARY"
For I = 13 To 17
txtFields(I).Text = "0"
Next
End If
data1.Recordset.Update
```

```
Else: data1.Recordset.CancelUpdate
End If
Call CONSTAT(Me)
End Sub

Private Sub CMDFIRST_Click()
data1.Recordset.MoveFirst
Call FIRSTAT(Me)
End Sub

Private Sub CMDLAST_Click()
data1.Recordset.MoveLast
Call LASTAT(Me)
End Sub

Private Sub CMDMAIN_Click()
FRMMAIN.Show
Unload Me
End Sub

Private Sub CMDNEXT_Click()
data1.Recordset.MoveNext
If data1.Recordset.EOF Then
data1.Recordset.MovePrevious
CMDNEXT.Enabled = False
CMDLAST.Enabled = False
MsgBox (" This Is The Last Record ")
Else
Call NEXSTAT(Me)
End If
End Sub

Private Sub CMDPREVIOUS_Click()
data1.Recordset.MovePrevious
If data1.Recordset.BOF Then
data1.Recordset.MoveNext
CMDFIRST.Enabled = False
CMDPREV.Enabled = False
MsgBox (" This Is The First Record ")
Else
Call PRESTAT(Me)
End If
End Sub
```

```
Private Sub CMDREMOVE_Click()
Dim VAL As Integer
VAL = MsgBox(" Are You Sure You Want To Remove This Record ", vbYesNo +
vbCritical)
If VAL = vbYes Then
data1.Recordset.Delete
data1.Recordset.MoveNext
If data1.Recordset.EOF Then
data1.Recordset.MoveLast
End If
End If
End Sub


Private Sub CMDSEARCH_Click()
Dim VAL As String, FIND As String, I As Integer
Dim POS As Integer
POS = data1.Recordset.AbsolutePosition
LINE1: VAL = UCase(Trim(InputBox(" Enter The Customer Code To Be Searched ")))
If VAL = "" Then
MsgBox (" There Should Be Some Value To Search ")
GoTo LINE1
Else: FIND = "CODE='" & VAL & "'"
data1.Recordset.FindFirst FIND
If data1.Recordset.NoMatch Then
data1.Recordset.AbsolutePosition = POS
I = MsgBox(" Record Not Found ", vbOKOnly + vbExclamation)
End If
Call SERSTAT(Me)
End If
End Sub

Private Sub CMDUPDATE_Click()
Dim V As Integer
txtFields(1).SetFocus
Call MODSTAT(Me)
V = MsgBox(" Press CONFIRM After Modification ", vbInformation)
data1.Recordset.Edit
STATUS = True
End Sub

Private Sub DATA1_Error(DataErr As Integer, Response As Integer)
  MsgBox "Data error event hit err:" & Error$(DataErr)
```

```
 Response = 0
 End Sub

Private Sub DATA1_Reposition()
 Screen.MousePointer = vbDefault
 On Error Resume Next
 data1.Caption = "Record: " & (data1.Recordset.AbsolutePosition + 1)
 If data1.Recordset.Fields(11) = "TEMPORARY" Then
    Option1.Value = True
    Call Option1_Click
    Else: Option2.Value = True
    Call Option2_Click
    End If
 End Sub

Private Sub DATA1_Validate(Action As Integer, Save As Integer)
 Select Case Action
  Case vbDataActionMoveFirst
  Case vbDataActionMovePrevious
  Case vbDataActionMoveNext
  Case vbDataActionMoveLast
  Case vbDataActionAddNew
  Case vbDataActionUpdate
  Case vbDataActionDelete
  Case vbDataActionFind
  Case vbDataActionBookmark
  Case vbDataActionClose
    Screen.MousePointer = vbDefault

 End Select
 Screen.MousePointer = vbHourglass
End Sub

Private Sub Form_Initialize()
CMDFIRST_Click
End Sub


Private Sub Form_Unload(Cancel As Integer)
 Screen.MousePointer = vbDefault
 FRMMAIN.Show
End Sub

Private Sub Option1_Click()
```

```
Dim I As Integer
If Option1.Value = True Then
For I = 13 To 17
lblLabels(I).Enabled = False
txtFields(I).Enabled = False
Next
If STATUS Then
For I = 13 To 17
txtFields(I).Text = "0"
Next
End If
End If
End Sub


Private Sub Option2_Click()
Dim I As Integer
If Option2.Value = True Then
For I = 13 To 17
lblLabels(I).Enabled = True
txtFields(I).Enabled = True
Next
End If
End Sub
```

CODE FOR FRMPURCH_ORD

```
Dim COUNTDET As Integer, STATUS As Boolean, UPSTAT As Boolean
Dim COUNT1 As Integer
Private Sub CLEAR()
Dim I As Integer
For I = 0 To 10
txtrat(I).Text = ""
txtqty(I).Text = ""
Combo(I).Text = ""
If I <> 0 Then
txtrat(I).Visible = False
txtqty(I).Visible = False
txtsl(I).Visible = False
Combo(I).Visible = False
End If
Next
End Sub
Private Sub CMDADD_Click()
Dim I As Integer, VAL1 As Integer, X As Integer
Dim STR As String
Call ADDSTAT(Me)
Data1.Recordset.AddNew
I = MsgBox(" Press 'ENTER' After Entering Data ", vbInformation)
STATUS = True
Data3.RecordSource = "SELECT MAX([NUM]) FROM [PUR_ORD_HEAD]"
Data3.Refresh
If IsNull(Data3.Recordset(0)) Then
TEXT1(0) = 1
Else
TEXT1(0) = Data3.Recordset(0) + 1
End If
Combo2.SetFocus
TEXT1(3) = Format(Now, "DD/MM/YYYY")
Call CLEAR
CMDCONFIRM.Default = True
End Sub

Private Sub CMDCONFIRM_Click()
Dim VAL1 As Integer, I As Integer, X As Integer
Dim STR As String
```

```
VAL1 = MsgBox(" Are You Sure You Want To Update This Record ", vbYesNo +
vbQuestion)
If VAL1 = vbYes Then
LORD:  If TEXT1(0).Text = "" Then
      STR = InputBox(" 'Order No' Should Contain A Value , Enter A Value", vbCritical)
      TEXT1(0).Text = STR
      GoTo LORD
      End If
LQUOT:       If Combo2.Text = "" Then
      MsgBox (" 'Quotation No' Should Contain A Value , Select A Value")
      GoTo LQUOT
      End If
LSUP:       If Combo1.Text = "" Then
      MsgBox (" 'Supplier Code' Should Contain A Value , Select A Value")
      GoTo LSUP
      End If
      For I = 4 To 6
      If I <> 5 Then
LINE1:       If Not IsNumeric(TEXT1(I)) Then
        STR = InputBox(LAB1(I).Caption + " Should Contain A Numeric Value , Please
Enter One ")
      TEXT1(I).Text = STR
      GoTo LINE1
      End If
      End If
      Next

      CMDREMOVE.Enabled = True
            If STATUS Then
            TEXT1(5) = 0
            Data2.RecordSource = "SELECT * FROM PUR_ORD_DET "
            Data2.Refresh
            For I = 0 To COUNT1
            Data2.Recordset.AddNew
            Data2.Recordset(0) = TEXT1(0)
            Data2.Recordset(4) = txtrat(I)
            Data2.Recordset(1) = txtsl(I)
            Data2.Recordset(3) = txtqty(I)
            Data2.Recordset(2) = Combo(I).Text
            TEXT1(5) = VAL(TEXT1(5)) + VAL(txtrat(I)) * VAL(txtqty(I))
            Data2.Recordset.Update
            Next
              TEXT1(5) = VAL(TEXT1(5)) + (VAL(TEXT1(5)) * VAL(TEXT1(4)) -
VAL(TEXT1(5)) * VAL(TEXT1(6))) / 100
```

```
            End If
        If UPSTAT Then
        TEXT1(5) = 0
         Data2.RecordSource = "SELECT * FROM PUR_ORD_DET WHERE NUM='"
& Data1.Recordset(0) & "' ORDER BY SNO"
        Data2.Refresh
        For I = 0 To Data2.Recordset.RecordCount - 1
        Data2.Recordset.Edit
            Data2.Recordset(0) = Data1.Recordset(0)
            Data2.Recordset(4) = txtrat(I)
            Data2.Recordset(1) = txtsl(I)
            Data2.Recordset(3) = txtqty(I)
            Data2.Recordset(2) = Combo(I).Text
            TEXT1(5) = VAL(TEXT1(5)) + VAL(txtrat(I)) * VAL(txtqty(I))
            Data2.Recordset.Update
            Data2.Recordset.MoveNext
        Next
        For I = Data2.Recordset.RecordCount To COUNT1 + 1
        Data2.Recordset.AddNew
            Data2.Recordset(0) = TEXT1(0)
            Data2.Recordset(4) = txtrat(I)
            Data2.Recordset(1) = txtsl(I)
            Data2.Recordset(3) = txtqty(I)
            Data2.Recordset(2) = Combo(I).Text
            TEXT1(5) = VAL(TEXT1(5)) + VAL(txtrat(I)) * VAL(txtqty(I))
            Data2.Recordset.Update
            Next
             TEXT1(5) = VAL(TEXT1(5)) + (VAL(TEXT1(5)) * VAL(TEXT1(4)) -
VAL(TEXT1(5)) * VAL(TEXT1(6))) / 100
        End If

Data1.Recordset.Update
If STATUS Then Data1.Recordset.MoveLast
Else
Data1.Recordset.CancelUpdate
End If
Call CONSTAT(Me)
STATUS = False
UPSTAT = False
CMDCONFIRM.Default = False
End Sub

Private Sub CMDFIRST_Click()
Data1.Recordset.MoveFirst
```

```
Call FIRSTAT(Me)
End Sub

Private Sub CMDLAST_Click()
Data1.Recordset.MoveLast
Call LASTAT(Me)
End Sub

Private Sub CMDMAIN_Click()
FRMMAIN.Show
Unload Me
End Sub

Private Sub CMDNEXT_Click()
Data1.Recordset.MoveNext
If Data1.Recordset.EOF Then
Data1.Recordset.MovePrevious
CMDNEXT.Enabled = False
CMDLAST.Enabled = False
MsgBox (" This Is The Last Record ")
Else
Call NEXSTAT(Me)
End If
End Sub

Private Sub CMDPREVIOUS_Click()
Data1.Recordset.MovePrevious
If Data1.Recordset.BOF Then
Data1.Recordset.MoveNext
CMDFIRST.Enabled = False
CMDPREVIOUS.Enabled = False
MsgBox (" This Is The First Record ")
Else
Call PRESTAT(Me)
End If
End Sub

Private Sub CMDREMOVE_Click()
Dim VAL1 As Integer, I As Integer
VAL1 = MsgBox(" Are You Sure You Want To Remove This Order ", vbYesNo +
vbCritical)
If VAL1 = vbYes Then

    Data2.Refresh
```

```
    While Not Data2.Recordset.EOF
    Data2.Recordset.Delete
    Data2.Recordset.MoveNext
    Wend
  Data1.Recordset.Delete
  Data1.Recordset.MovePrevious
   If Data1.Recordset.BOF Then
   CMDREMOVE.Enabled = False
   End If
End If
End Sub

Private Sub CMDSEARCH_Click()
Dim VAL1 As String, FIND As String, I As Integer
Dim POS As Integer
POS = Data1.Recordset.AbsolutePosition
LINE1: VAL1 = UCase(Trim(InputBox(" Enter The Order Number To Be Searched ")))
If VAL1 = "" Then
MsgBox (" There Should Be Some Value To Search ")
GoTo LINE1
Else: FIND = "NUM='" & VAL1 & "'"
Data1.Recordset.FindFirst FIND
If Data1.Recordset.NoMatch Then
Data1.Recordset.AbsolutePosition = POS
I = MsgBox(" Record Not Found ", vbOKOnly + vbExclamation)
End If
Call SERSTAT(Me)
End If
End Sub

Private Sub CMDUPDATE_Click()
Dim V As Integer
TEXT1(0).SetFocus
Call MODSTAT(Me)
V = MsgBox(" Press 'ENTER' After Modification ", vbInformation)
Data1.Recordset.Edit
UPSTAT = True
CMDCONFIRM.Default = True
End Sub

Private Sub Combo_GotFocus(Index As Integer)
Dim I As Integer
If STATUS Or UPSTAT Then
Combo(Index).CLEAR
```

```
Data3.RecordSource = "SELECT PART_CODE FROM PART_MAST"
Data3.Refresh
For I = 0 To Data3.Recordset.RecordCount
   Combo(Index).AddItem Data3.Recordset(0)
   Data3.Recordset.MoveNext
Next
End If
End Sub


Private Sub Combo1_GotFocus()
If STATUS Or UPSTAT Then
Data3.RecordSource = "SELECT CODE FROM SUPPLY_MAST ORDER BY CODE"
Data3.Refresh
Combo1.CLEAR
While Not Data3.Recordset.EOF
Combo1.AddItem Data3.Recordset(0)
Data3.Recordset.MoveNext
Wend
End If
End Sub


Private Sub Combo2_GotFocus()
If STATUS Or UPSTAT Then
Data3.RecordSource = "SELECT QUOTATION_NO FROM QUOTATION_HEAD
ORDER BY QUOTATION_NO"
Data3.Refresh
Combo2.CLEAR
While Not Data3.Recordset.EOF
Combo2.AddItem Data3.Recordset(0)
Data3.Recordset.MoveNext
Wend
End If
End Sub

Private Sub DATA1_Reposition()
Dim I As Integer
Call CLEAR
If Not (Data1.Recordset.EOF Or Data1.Recordset.BOF) Then
Data2.RecordSource = "SELECT * FROM PUR_ORD_DET WHERE NUM = '" &
Data1.Recordset(0) & "' ORDER BY SNO"
Data2.Refresh
For I = 0 To Data2.Recordset.RecordCount - 1
```

```
If I <> 0 Then
txtsl(I).Visible = True
txtrat(I).Visible = True
Combo(I).Visible = True
txtqty(I).Visible = True
End If
txtrat(I) = Data2.Recordset(4)
txtqty(I) = Data2.Recordset(3)
Combo(I).Text = Data2.Recordset(2)
Data2.Recordset.MoveNext
Next
End If
End Sub

Private Sub Form_Activate()
SSTab1.Tab = 0
End Sub

Private Sub Form_Unload(Cancel As Integer)
FRMMAIN.Show
End Sub

Private Sub TEXT1_GotFocus(Index As Integer)
If Index = 5 Then CMDCONFIRM.Default = True
End Sub

Private Sub TEXT1_LostFocus(Index As Integer)
If Index = 6 Then
SSTab1.Tab = 1
Combo(0).SetFocus
End If
End Sub

Private Sub txtqty_LostFocus(Index As Integer)
If STATUS Or (UPSTAT And Not txtsl(Index + 1).Visible) Then
    If (MsgBox("Do You Want To Add More Parts", vbYesNo + vbq) = vbYes) And
Index <> 11 Then
   txtsl(Index + 1).Visible = True
   txtrat(Index + 1).Visible = True
   txtqty(Index + 1).Visible = True
   Combo(Index + 1).Visible = True
   Combo(Index + 1).SetFocus
   Else
   COUNT1 = Index
```

```
    SSTab1.Tab = 0
    TEXT1(7).SetFocus
End If
End If
End Sub


CODE FOR FRMSPARES_ARRIVAL:
Dim STAT As Boolean, UPSTAT As Boolean
Private Sub CMDADD_Click()
Dim I As Integer, VAL1 As Integer, X As Integer
Call ADDSTAT(Me)
data1.Recordset.AddNew
I = MsgBox(" Press 'CONFIRM' After Entering Data ", vbInformation)
Data2.RecordSource = "SELECT MAX(PARNO) FROM PARTS_ARRIVAL "
Data2.Refresh
If IsNull(Data2.Recordset(0)) Then
txtFields(0) = 1
Else
txtFields(0) = (Data2.Recordset(0) + 1)
End If
CMDCONFIRM.Default = True
STAT = True
txtFields(2) = Format(Now, "DD/MM/YYYY")
Combo1.SetFocus
End Sub

Private Sub CMDCONFIRM_Click()
Dim VAL1 As Integer, I As Integer, X As Integer
Dim STR As String
VAL1 = MsgBox(" Are You Sure You Want To Update This Record ", vbYesNo +
vbQuestion)
If VAL1 = vbYes Then
LPAR: If txtFields(0).Text = "" Then
STR = InputBox(" 'PAR_NO' Should Contain A Value , Enter A Value")
txtFields(0).Text = STR
GoTo LPAR
End If
LSUP: If Combo2.Text = "" Then
MsgBox (" 'SUP CODE' Should Contain A Value , Select A Value")
GoTo LSUP
End If
LPART: If Combo1.Text = "" Then
MsgBox (" 'PART CODE' Should Contain A Value , Select A Value")
```

```
GoTo LPART
End If
LINE1: If Not IsNumeric(txtFields(4)) Then
STR = InputBox(lblLabels(4).Caption + " Should Contain A Numeric Value , Please
Enter One ")
txtFields(4).Text = STR
GoTo LINE1
End If
CMDREMOVE.Enabled = True
                Data2.RecordSource = "SELECT * FROM PART_MAST WHERE
PART_CODE ='" & Combo1.Text & "'"
        Data2.Refresh
        Data2.Recordset.Edit
        Data2.Recordset(2) = Data2.Recordset(2) + VAL(txtFields(4))
        If UPSTAT Then
        Data2.Recordset(2) = Data2.Recordset(2) - data1.Recordset(4)
        End If
        Data2.Recordset.Update
data1.Recordset.Update
data1.Recordset.MoveLast
Else
data1.Recordset.CancelUpdate
End If
Call CONSTAT(Me)
CMDCONFIRM.Default = False
UPSTAT = False
STAT = False
End Sub

Private Sub CMDFIRST_Click()
data1.Recordset.MoveFirst
Call FIRSTAT(Me)
End Sub

Private Sub CMDLAST_Click()
data1.Recordset.MoveLast
Call LASTAT(Me)
End Sub

Private Sub CMDMAIN_Click()
FRMMAIN.Show
Unload Me
End Sub
```


```
Private Sub CMDNEXT_Click()
data1.Recordset.MoveNext
If data1.Recordset.EOF Then
data1.Recordset.MovePrevious
CMDNEXT.Enabled = False
CMDLAST.Enabled = False
MsgBox (" This Is The Last Record ")
Else
Call NEXSTAT(Me)
End If
End Sub

Private Sub CMDPREVIOUS_Click()
data1.Recordset.MovePrevious
If data1.Recordset.BOF Then
data1.Recordset.MoveNext
CMDFIRST.Enabled = False
CMDPREVIOUS.Enabled = False
MsgBox (" This Is The First Record ")
Else
Call PRESTAT(Me)
End If
End Sub

Private Sub CMDREMOVE_Click()
Dim VAL1 As Integer
VAL1 = MsgBox(" Are You Sure You Want To Remove This Record ", vbYesNo +
vbCritical)
If VAL1 = vbYes Then
data1.Recordset.Delete
data1.Recordset.MovePrevious
If data1.Recordset.BOF Then
data1.Refresh
CMDREMOVE.Enabled = False
End If
End If
End Sub

Private Sub CMDSEARCH_Click()
Dim VAL1 As String, FIND As String, I As Integer
Dim POS As Integer
POS = data1.Recordset.AbsolutePosition
LINE1: VAL1 = UCase(Trim(InputBox(" Enter The P.A.R NO To Be Searched ")))
If VAL1 = "" Then
```

```
MsgBox (" There Should Be Some Value To Search ")
GoTo LINE1
Else: FIND = "PARNO='" & VAL1 & "'"
data1.Recordset.FindFirst FIND
If data1.Recordset.NoMatch Then
data1.Recordset.AbsolutePosition = POS
I = MsgBox(" Record Not Found ", vbOKOnly + vbExclamation)
End If
End If
Call SERSTAT(Me)
End Sub

Private Sub CMDUPDATE_Click()
Dim V As Integer
txtFields(0).SetFocus
Call MODSTAT(Me)
V = MsgBox(" Press CONFIRM After Modification ", vbInformation)
data1.Recordset.Edit
CMDCONFIRM.Default = True
UPSTAT = True
End Sub

Private Sub Combo1_GotFocus()
If STAT Or UPSTAT Then
   Combo1.CLEAR
    Data2.RecordSource = "SELECT PART_CODE FROM PART_MAST ORDER BY
PART_CODE"
   Data2.Refresh
   While Not Data2.Recordset.EOF
   Combo1.AddItem Data2.Recordset(0)
   Data2.Recordset.MoveNext
   Wend
End If
End Sub


Private Sub Combo2_GotFocus()
If STAT Or UPSTAT Then
Data2.RecordSource = "SELECT CODE FROM SUPPLY_MAST ORDER BY CODE"
Data2.Refresh
Combo2.CLEAR
While Not Data2.Recordset.EOF
Combo2.AddItem Data2.Recordset(0)
Data2.Recordset.MoveNext
```

```
Wend
End If
End Sub

Private Sub DATA1_Error(DataErr As Integer, Response As Integer)
 'This is where you would put error handling code
 'If you want to ignore errors, comment out the next line
 'If you want to trap them, add code here to handle them
 MsgBox "Data error event hit err:" & Error$(DataErr)
 Response = 0  'Throw away the error
End Sub


Private Sub DATA1_Validate(Action As Integer, Save As Integer)
 'This is where you put validation code
 'This event gets called when the following actions occur
 Select Case Action
  Case vbDataActionMoveFirst
  Case vbDataActionMovePrevious
  Case vbDataActionMoveNext
  Case vbDataActionMoveLast
  Case vbDataActionAddNew
  Case vbDataActionUpdate
  Case vbDataActionDelete
  Case vbDataActionFind
  Case vbDataActionBookmark
  Case vbDataActionClose
   Screen.MousePointer = vbDefault
 End Select

End Sub

Private Sub Form_Initialize()
data1.Refresh
CMDFIRST.Enabled = False
CMDPREVIOUS.Enabled = False
End Sub

Private Sub Form_Load()
Me.Picture = FRMMAIN.Picture
data1.Refresh
End Sub

Private Sub Form_Unload(Cancel As Integer)
```

```
  Screen.MousePointer = vbDefault
  FRMMAIN.Show
End Sub
```

CODE FOR FRMSLS_ENQ

```
Dim COUNTDET As Integer, STATUS As Boolean, UPSTAT As Boolean
Dim COUNT1 As Integer
Private Sub CLEAR()
Dim I As Integer
For I = 0 To 10
txtqty(I).Text = ""
Combo(I).Text = ""
If I <> 0 Then
txtqty(I).Visible = False
txtsl(I).Visible = False
Combo(I).Visible = False
End If
.Next
End Sub
Private Sub CMDADD_Click()
Dim I As Integer, VAL As Integer, X As Integer
Dim STR As String
Call ADDSTAT(Me)
Data1.Recordset.AddNew
I = MsgBox("          Press 'ENTER' After Entering Data          ", vbInformation)
STATUS = True
Data3.RecordSource = "SELECT MAX([QUERY_NO]) FROM [QUERY_HEAD]"
Data3.Refresh
If IsNull(Data1.Recordset(0)) Then
TEXT1(1) = 1
Else
TEXT1(1) = Data3.Recordset(0) + 1
End If
TEXT1(2) = Format(Now, "DD/MM/YYYY")
Combo1.SetFocus
Call CLEAR
End Sub

Private Sub CMDCONFIRM_Click()
Dim VAL As Integer, I As Integer, X As Integer
Dim STR As String
VAL = MsgBox("          Are You Sure You Want To Update This Record          ",
vbYesNo + vbQuestion)
If VAL = vbYes Then
```

```
LCODE: If Combo1.Text = "" Then
MsgBox ("       'CUS_CODE' Should Contain A Value , Select A Value          ")
GoTo LCODE
End If
LNO: If TEXT1(1) = "" Then
STR = InputBox("       'ENQUIRY_NO' Should Contain A Value , Enter A Value       ")
TEXT1(1) = STR
GoTo LNO
End If
LDAT: If TEXT1(2) = "" Then
STR = InputBox("       'Date' Should Contain A Valid Date , Please Enter One          ")
TEXT1(2) = STR
GoTo LDAT
End If
      CMDREMOVE.Enabled = True
            If STATUS Then
            Data2.RecordSource = "SELECT * FROM PUR_ORD_DET "
            Data2.Refresh
            For I = 0 To COUNT1
            Data2.Recordset.AddNew
            Data2.Recordset(0) = Combo1.Text
            Data2.Recordset(1) = txtsl(I)
            Data2.Recordset(3) = txtqty(I)
            Data2.Recordset(2) = Combo(I).Text
            Data2.Recordset.Update
            Next
            End If
       If UPSTAT Then
        Data2.RecordSource = "SELECT * FROM PUR_ORD_DET WHERE NUM='"
& Data1.Recordset(0) & "' ORDER BY SNO"
       Data2.Refresh
       For I = 0 To Data2.Recordset.RecordCount - 1
       Data2.Recordset.Edit
            Data2.Recordset(0) = Data1.Recordset(0)
            Data2.Recordset(1) = txtsl(I)
            Data2.Recordset(3) = txtqty(I)
            Data2.Recordset(2) = Combo(I).Text
            Data2.Recordset.Update
            Data2.Recordset.MoveNext
       Next
       For I = Data2.Recordset.RecordCount To COUNT1 + 1
       Data2.Recordset.AddNew
            Data2.Recordset(0) = Combo1.Text
            Data2.Recordset(1) = txtsl(I)
```

```
                Data2.Recordset(3) = txtqty(I)
                Data2.Recordset(2) = Combo(I).Text
                Data2.Recordset.Update
                Next
            End If

    Data1.Recordset.Update
    If STATUS Then Data1.Recordset.MoveLast
    Else
    Data1.Recordset.CancelUpdate
    End If
    Call CONSTAT(Me)
    STATUS = False
    UPSTAT = False
    CMDCONFIRM.Default = False
    End Sub

    Private Sub CMDFIRST_Click()
    Data1.Recordset.MoveFirst
    Call FIRSTAT(Me)
    End Sub

    Private Sub CMDLAST_Click()
    Data1.Recordset.MoveLast
    Call LASTAT(Me)
    End Sub

    Private Sub CMDMAIN_Click()
    FRMMAIN.Show
    Unload Me
    End Sub

    Private Sub CMDNEXT_Click()
    Data1.Recordset.MoveNext
    If Data1.Recordset.EOF Then
    Data1.Recordset.MovePrevious
    CMDNEXT.Enabled = False
    CMDLAST.Enabled = False
    MsgBox ("       This Is The Last Record       ")
    Else
    Call NEXSTAT(Me)
    End If
    End Sub
```

```
Private Sub CMDPREVIOUS_Click()
Data1.Recordset.MovePrevious
If Data1.Recordset.BOF Then
Data1.Recordset.MoveNext
CMDFIRST.Enabled = False
CMDPREVIOUS.Enabled = False
MsgBox ("       This Is The First Record       ")
Else
Call PRESTAT(Me)
End If
End Sub

Private Sub CMDREMOVE_Click()
Dim VAL As Integer
VAL = MsgBox(" Are You Sure You Want To Remove This Record ", vbYesNo +
vbCritical)
If VAL = vbYes Then
    Data2.Recordset.MoveFirst
    While Not Data2.Recordset.EOF
    Data2.Recordset.Delete
    Data2.Recordset.MoveNext
    Wend
Data1.Recordset.Delete
Data1.Recordset.MovePrevious
If Data1.Recordset.BOF Then
CMDREMOVE.Enabled = False
End If
End If
End Sub

Private Sub CMDSEARCH_Click()
Dim VAL As String, FIND As String, I As Integer
Dim POS As Integer
POS = Data1.Recordset.AbsolutePosition
LINE1: VAL = UCase(Trim(InputBox(" Enter The Enquiry Number To Be Searched ")))
If VAL = "" Then
MsgBox (" There Should Be Some Value To Search ")
GoTo LINE1
Else
FIND = "QUERY_NO='" & VAL & "'"
Data1.Recordset.FindFirst FIND
If Data1.Recordset.NoMatch Then
Data1.Recordset.AbsolutePosition = POS
I = MsgBox("       Record Not Found       ", vbOKOnly + vbExclamation)
```

```
End If
Call SERSTAT(Me)
End If
End Sub

Private Sub CMDUPDATE_Click()
Dim V As Integer
Combo1.SetFocus
Call MODSTAT(Me)
V = MsgBox("          Press CONFIRM After Modification          ", vbInformation)
Data1.Recordset.Edit
UPSTAT = True
CMDCONFIRM.Default = True
End Sub

Private Sub Combo_GotFocus(Index As Integer)
Dim I As Integer
If STATUS Then
Combo(Index).CLEAR
Data3.RecordSource = "SELECT PROD_CODE FROM PRODUCT_MAST"
Data3.Refresh
While Not Data3.Recordset.EOF
   Combo(Index).AddItem Data3.Recordset(0)
   Data3.Recordset.MoveNext
Wend
End If
End Sub


Private Sub Combo1_GotFocus()
If STATUS Or UPSTAT Then
Data3.RecordSource = "SELECT CODE FROM CUSTOMER ORDER BY CODE"
Data3.Refresh
Combo1.CLEAR
While Not Data3.Recordset.EOF
Combo1.AddItem Data3.Recordset(0)
Data3.Recordset.MoveNext
Wend
End If
End Sub

Private Sub DATA1_Reposition()
Dim I As Integer
Call CLEAR
```

```
If Not (Data1.Recordset.EOF Or Data1.Recordset.BOF) Then
Data2.RecordSource = "SELECT * FROM QUERY_DET WHERE QUERY_NO = '" &
Data1.Recordset(2) & "' ORDER BY SNO"
Data2.Refresh
For I = 0 To Data2.Recordset.RecordCount - 1

If I <> 0 Then
txtsl(I).Visible = True
Combo(I).Visible = True
txtqty(I).Visible = True
End If

txtqty(I) = Data2.Recordset(2)
Combo(I).Text = Data2.Recordset(1)
Data2.Recordset.MoveNext
Next
End If

End Sub

Private Sub Form_Activate()
SSTab1.Tab = 0
End Sub

Private Sub Form_Unload(Cancel As Integer)
FRMMAIN.Show
If STATUS Or UPSAT Then
Data1.Recordset.CancelUpdate
End If
End Sub

Private Sub TEXT1_LostFocus(Index As Integer)
If Index = 2 Then
SSTab1.Tab = 1
Combo(0).SetFocus
End If
End Sub

Private Sub txtqty_LostFocus(Index As Integer)
```

SCREENS

REPORTS