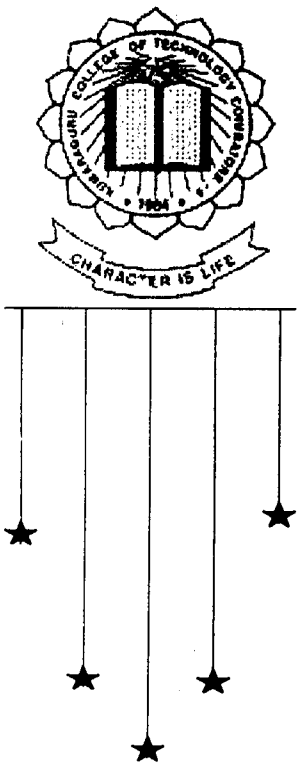
MAINTENANCE, MANAGEMENT AND FINANCIAL ACCOUNTING OF A DEPARTMENTAL STORES



PROJECT REPORT

Submitted By

MEYYAPPAN .S

SATHISH ARUN .S.P

REKHA .M

Under the Guidance of

Mr.M. SARAVANA MUTHU, M.C.A.,

In partial fulfillment of the requirements for the award of the degree of

BACHELOR OF SCIENCE IN 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 – 641006.

DEPARTMENT OF COMPUTER SCIENCE AND TECHNOLOGY

Certificate

This is to certify that the project entitled

MAINTENANCE, MANAGEMENT AND FINANCIAL ACCOUNTING OF A DEPARTMENTAL STORES

has been submitted by

Mr/~~Ms.~~ MEYYAPPAN.S , Ms.REKHA.M , SATHISHARUN.S.P

in partial fulfillment of the requirements for the award of Degree of

BACHELOR OF SCIENCE IN

APPLIED SCIENCE – COMPUTER TECHNOLOGY

Branch of the Bharathiar University, Coimbatore,

During the academic year 2000-2001.

14/3/2001
Guide

14/3/'01
Head of the Department

Certified that the candidate with register number 7Q0134,9827Q0155,9827Q0160 was examined in the project work and viva-voce Examination held on 16-03-2001

16/03
Internal Examiner

16/3/2001
External Examiner

IRA DEPARTMENTAL STORES
Sale and Retail Traders

544, Trichy Road,
Singanallur,
Coimbatore – 641 005.
Phone: 572880, 592936.

TO WHOMSOEVER IT MAY CONCERN

This is to certify that **Mr.Meyyappan.S, Mr.Sathish Arun S.P., Ms.Rekha .M,** have developed a **"MAINTENANCE, MANAGEMENT AND FINANCIAL ACCOUNTING"** package for our company. This deals with the whole computerisation of the departmental stores. They developed this package as trainees and we agreed to implement the package done by them in our departmental stores.

We are very much happy with the software, and the interest they showed on the project.

We wish them all success in their career.

With regards,

S. Srinivasan

(S. SRINIVASAN)
Proprietor.

DECLARATION

I hereby declare that this project entitled "MAINTENANCE, MANAGEMENT AND FINANCIAL ACCOUNTING OF A DEPARTMENTAL STORES" submitted is a record of original work done by me under the supervision of Mr.,M.Saravana Muthu, Lecturer, Department of Computer Technology and Application, Kumaraguru College of Technology and that this project work as not formed the basis for the award of any degree / diploma / associateship / fellowship with similar title with any other University.

Place: Coimbatore.
Date :

ENDORSED BY:

Mr.M.Saravana Muthu

Place: Coimbatore
Date :

CONTENTS

ACKNOWLEDGEMENT

We wish to express our regards and sincere thanks to the management of Kumaraguru College of Technology and to our Principal K.K.Padmanabhan for providing necessary facilities to carry out this project work.

We feel highly elated in manifesting our deep sense of thankfulness to our Head of the Department Prof.S.thangasamy and our Guide Mr.M. Saravana Muthu for their keen interest, valuable guidance, useful suggestions and constant help during the course of this project work. We can say with utmost confidence that they were the backbone of this project. We are thankful to our staffs, parents and our friends for the completion of the project.

Though the words are not enough, it is all that we have got to express deepest gratitude to Mr.S.Srinivasan the Proprietor of Indra Departmental Stores, Coimbatore for readily providing all the facilities, patiently answering all our queries and helping us to carry out this project meticulously.

1. INTRODUCTION

1.1. PURPOSE

The main aim of this project is to computerize the manual operations in a Departmental Stores. This we have undertaken for the computerization of the manual transactions in the stores.

Due to the concept of employing computers, lots of tensions and the physical stresses are reduced a lot. There is no need to pay more attention to all the work in the stores. The great difficulty in working will be decreased a lot as we have employed the computers .

1.2. SCOPE

The scope of this project is that it helps in reducing the daily tension undergone by the people in the stores. It makes the people get out of the fatigue. The computerised system is implemented in easy way than the manual system.

The scope of the project is that it helps the people from the physical and the mental stresses due to the heavy and difficult job.

1.3. ACRONYMNS

DLL	-	Dynamic Linking Library
BMP	-	Bit Map Image
VBP	-	Visual Basic Project
MDB	-	Microsoft Data Base
JPEG	-	Joint Picture Editing Group
FRM	-	Form

1.4. REFERENCES:

Books:

Software Engineering - Roger S.Pressman

VB 6.0 - Peter Wrights

Beginning VB 6.0 - Gary Cornell

Microsoft Access 7

An integrated approach to

Software engineering - Pankaj Jalote

People:

The people in the stores who have helped us a lot being with us.

Mr.S.Srinivasan - Managing Director

Mr.S.Kumars Assistant Manager

1.5. OVERVIEW

The project is based on the making of the manual Departmental Stores into the computerized one. The quality of the work done by the machine is usually uniform, neat and reliable than when it is done manually. The operating cost is very high while running this manually. The operating cost is lowered by employing computers in this field. The efficiency and accuracy.

Is high in employing computers. The need for the computerization is for the quick job completion and the updation and the deletions can be performed easily.

Our project aims to computerize the manually operating Departmental stores. Previously, the stores was running manually which was a quite difficult job undertaken. Used more workmen in order to satisfy the customer's needs.

Nowadays, the number of people has been increased to a lot and hence their needs have to be fulfilled in one way or other. We are in a position to satisfy these persons and also get ourselves satisfied all our needs. Limited number of persons are not enough out everywhere. Hence all the persons are to be posted a job for everyone's well being.

Employing computers will make the business run in a smooth way. Even this computerization reduces the mental and physical stresses of the people. This concept of employing computers will bring the great relief of the people from the mental torture and fatigueness. This concept doesn't require more people on hand and easy to adopt. All the details such as the sales, stock, purchase, customer and license details are clearly shown in the screen. Even the Income tax dues are also shown exactly period wise.

We have coded three modules like Maintenance, Management and Financial Accounting of the stores. In the first module Maintenance, the stock of the products in the stores are maintained and are checked for the full amount of it. If the reorder level is maintained in a serial way then the stock is full, otherwise the amount of the stock has been reduced. The particular stock has to be purchased.

In the second module Management, the details of License ,Income Tax are maintained. The customer details are stored in the system and whenever any need , the customer can be intimated directly. Through this a good relationship is maintained between the people and the customers. The purchase details are also stored in this which makes the cash bill to be generated.

Finally, the third module Financial Accounting prepares the bills for the purchases made and the balance sheet is generated for that. Ultimately, the reports are generated for the bills.

This is to compute the yearly turn-over of the stores.

2. SYSTEM ANALYSIS

2.1 NEED FOR COMPUTERIZATION

Computerization has become an important part of the modern administrative process.

The quality of the work performed by a machine is usually uniform, neat and more reliable than when it is done manually. The operating cost of running all these works is high. The operating cost is lowered by employing computers in this field .The efficiency and accuracy is also increased in employing computers.

The benefit of computerization is quick job completion. The updation and deletion can also be performed quickly and accurately.

2.2 PROPOSED SYSTEM

Our project is a work-station based project. These are stand-alone systems .Hence these are not needed to connect to the server or to the network. Any person can access the system at any place.

This facility is provided as we use the platforms Visual Basic 6.0. and Microsoft Access.

Using Visual Basic we can satisfy any great needs of the users at the earliest possible time. It is quite easy to implement these functions in the project without any difficulty.

This language is easy to read and understand and hence quite interesting to work with. This is similar to the Basic language by which a single command invokes most of all functions easily.

Microsoft Access is similar to the Oracle language which serves the best in the storage of data. Creating tables is an easy task and hence the first. Our project has three modules like

a). Maintenance

b). Management

c). Financial Accounting

2.3 HARDWARE SPECIFICATION

Processor	:	Pentium
RAM	:	64 mb
Hard disk capacity	:	4.2 GB

Serial Mouse	:	MS compatible Mouse
Printer	:	HP Laser-Jet
Ports	:	2 serial ports , 1 parallel port.

2.4 SOFTWARE SPECIFICATION:

Platforms used	:	Windows NT server 4.0 /
		Windows 95
		Windows 98
Front End Tool	:	Visual Basic 6.0
Back End Tool	:	Microsoft Access
Reporting Tool	:	Data Report
Components Used	:	ActiveX Data Objects

3. SYSTEM DESIGN

3.1. INPUT DESIGN:

The Input Design describes the total number of input screens available. The designing is made of these input screens .Generally, the design process plays an important role in the project management process. The purpose of the design phase is to plan a solution of the problem specified by the requirements document. In other words, starting with *what* is needed , design takes us towards *how* to satisfy the needs.

The system design is sometimes called the Top-Level design, aims to identify the modules that should be in the system and how they interact with each other to produce the desired results.

We have designed five input screens totally, called as the Master Entries. Now let us see the five Master screens,

Master Entries:

1. Supplier Entry
2. License Entry
3. Customer Entry
4. Employee Entry
5. Item Entry.

Customer Entry:

In the customer entry the full details of the customers in the shop are stored in the screen format. The customer's name, age, sex, phone number, fax number, address, occupation are all stored in the screen. These details of the customers are available to the stores owners. For the purpose of door-delivery of products.

Also whenever any customer forgets to purchase any regular items or products, the owner of the stores could remind the regular customers using either phone numbers or their addresses.

If suppose the customers forget to pay the amount for the purchases they made the people in the stores can very well remind the customers. For these purposes the details of the customers are stored.

Employee Entry:

In the employee entry the details regarding the workmen or the employees in the stores are maintained.

Simply, the salary, wages has to be given to the employees for their sincere work performed. During the festivals, bonus will be provided specially for the workmen. These will be stored in the employee entry.

Item Entry:

The items in the stores are maintained as by three levels like maximum level, minimum level, reorder level. The item code, name, cost price, selling price, quantity and the quality are stocked.

Scheduling:

The scheduling has two parts such as

a). Bill Delivery

b). Function Delivery.

These explain the delivering of the bills in which the advance amount and the total amount has to be recorded.

Financial Accounting:

This module has three divisions such as

a). Balance Sheet

b). Data Entry Details

c). Text Editor

Balance sheet:

The salary , wages, bonus, sundry creditors, loans are filled in the input column of the balance sheet.

These are related to the employees and are given as inputs to the balance Sheet. The balance sheet is calculated by these information .

Data Entry Details:

The details are about the Electricity Board , LIC premium, House repair ,general expenses, vehicle maintenance , municipal tax charges.

Text Editor

Text Editor serves as the notepad functions .

3.2. TRANSACTION DESIGN

Here the details of the balance sheet generated are posted to financial accounting . The user assigns a code for the product as the *pcode.* The code of the product, *pcode* is assigned as the *primary key.*

The primary key is assigned for the purpose of the code to be the unique code. The code should not be repeated and the item name should also be repeated , only should be unique.

3.3 DATABASE DESIGN

The fields in the database design are based on the tables defined in the database. We have some 7 types of the database tables and only through this the data are entered.

Employee

Serial No.	Field Name	Field Type
1.	Name	text
2.	Age	integer
3.	Sex	boolean
4.	Addresss	text
5.	Contact number	integer
6.	Date of Joining	date
7.	Salary	integer
8.	Bonus	integer
9.	Department	text
10.	TotalPay	double

Customer

Serial No.	Field Name	Field Type
1.	Customer code	integer
2.	customer name	text
3.	street	text
4.	address	text
5.	city	text
6.	state	text
7.	zip	integer
8.	phone number	integer
9.	last transaction	date

Item

Serial No.	Field name	Field Type
1.	code	text
2.	name	text
3.	cost price	double
4.	selling price	double
5.	category	text
6.	unit	integer
7.	maximum quantity	integer
8.	minimum quantity	double
9.	reorder level	integer

License Details

Serial No.	Field Name	Field type
1.	name	text
2.	validation period	integer
3.	amount to be paid	integer
4.	date last paid	date
5.	date of updation	date

Cash Bill Generation

1. Input:

Serial No.	Field Name	Field Type
1.	name	text
2.	unit price	integer
3.	pd no.	integer

2. *Report*

Serial no.	Field name	field type
1.	pd no.	integer
2.	name	text
3.	qty	double
4.	unit price	double
5.	total	double

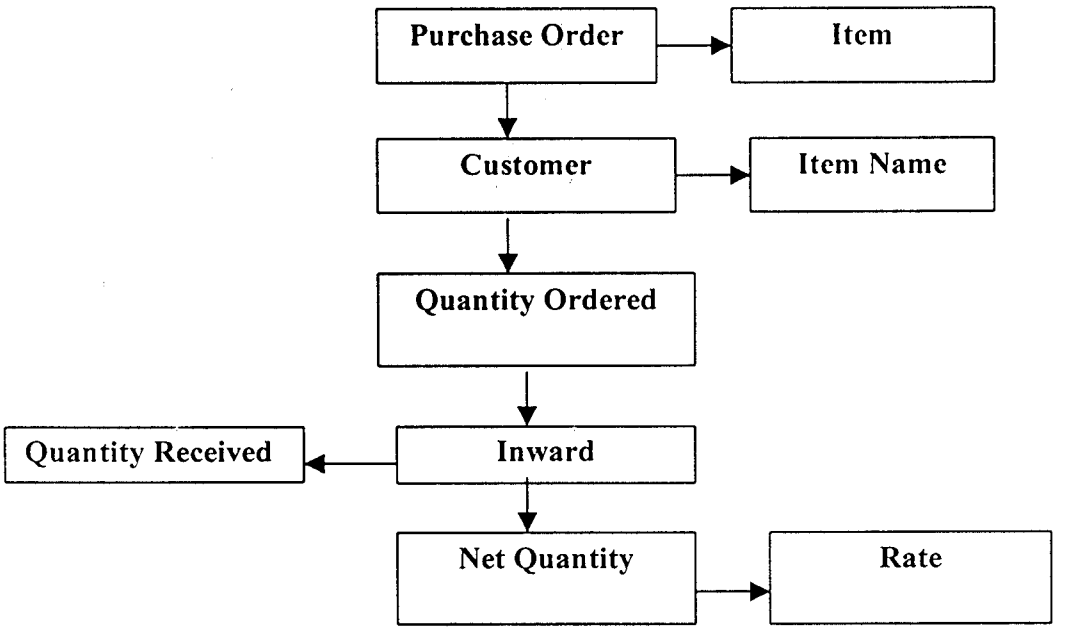
SCHEDULE

1. Bill

Serial no.	field name	field type
1.	name	text
2.	add	text
3.	date for delivery	date
4.	total amount	double
5.	place for delivery	text

4. SYSTEM FLOW DIAGRAM

MAINTENANCE



5. TESTING

Testing is the major quality control measure employed during software development. Its basic function is to detect errors in the software. During textual and non-executable. The goal of this testing is to uncover requirement, design or coding errors in the programs. Testing is a method that involves executing the code and is used . The different modules are tested separately.

5.1. UNIT TESTING

The starting point of the testing is the Unit testing. In this a module is tested separately and is often performed by the coder himself simultaneously with the coding of the module.

The purpose is to exercise the different parts of the module code to detect coding errors. After this the modules are gradually integrated into subsystems , which are often then integrated themselves to eventually form the entire system.

5.2. SYSTEM TESTING

In the system testing the system is tested against the system requirements to see if all the requirements are met and the system performs as specified by the requirements.

5.3. ACCEPTANCE TESTING

The Acceptance testing is performed to demonstrate to the client, on the real life data of the client, the operation of the system. For the testing to be successful , the proper selection of testcases are to be chosen. It is extremely critical and time consuming activity. It requires proper planning of the overall testing process.

This starts with a test plan. Then for different test units, a testcase specification document is produced, which lists all test cases, together with the expected outputs, that will be used for the testing.

6. CONCLUSION

All the three modules in our project works perfectly and is all set for the implementation of the various operations in the daily transactions of the departmental stores. This project have been very useful, informative and at the most educative. It has helped in understanding the practical problems in real life situation. The study conducted has been very useful to the stores as well as to us. It has given the completeness to the education received by us during the past three years of our degree course.

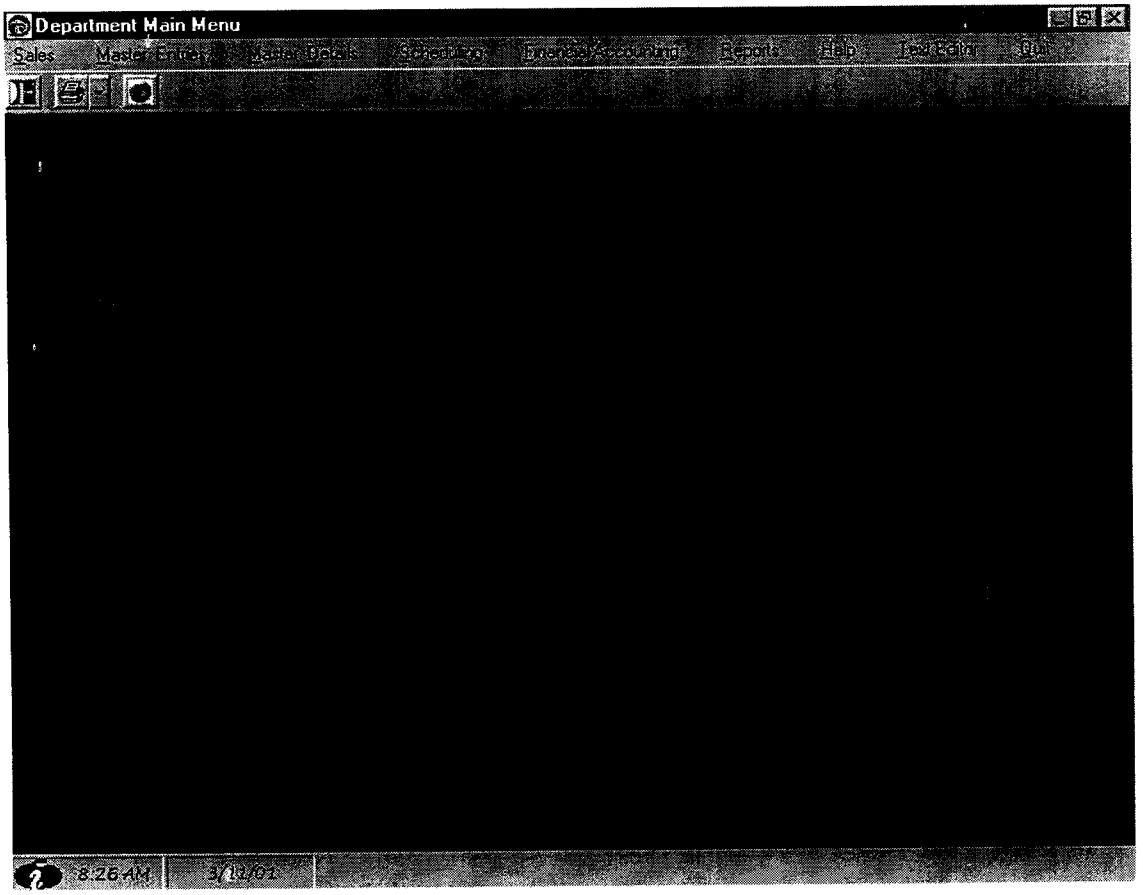
The system has been developed for the conditions existing at the present. The system being flexible , can be further enhanced as per the wishes of the user's requirements.

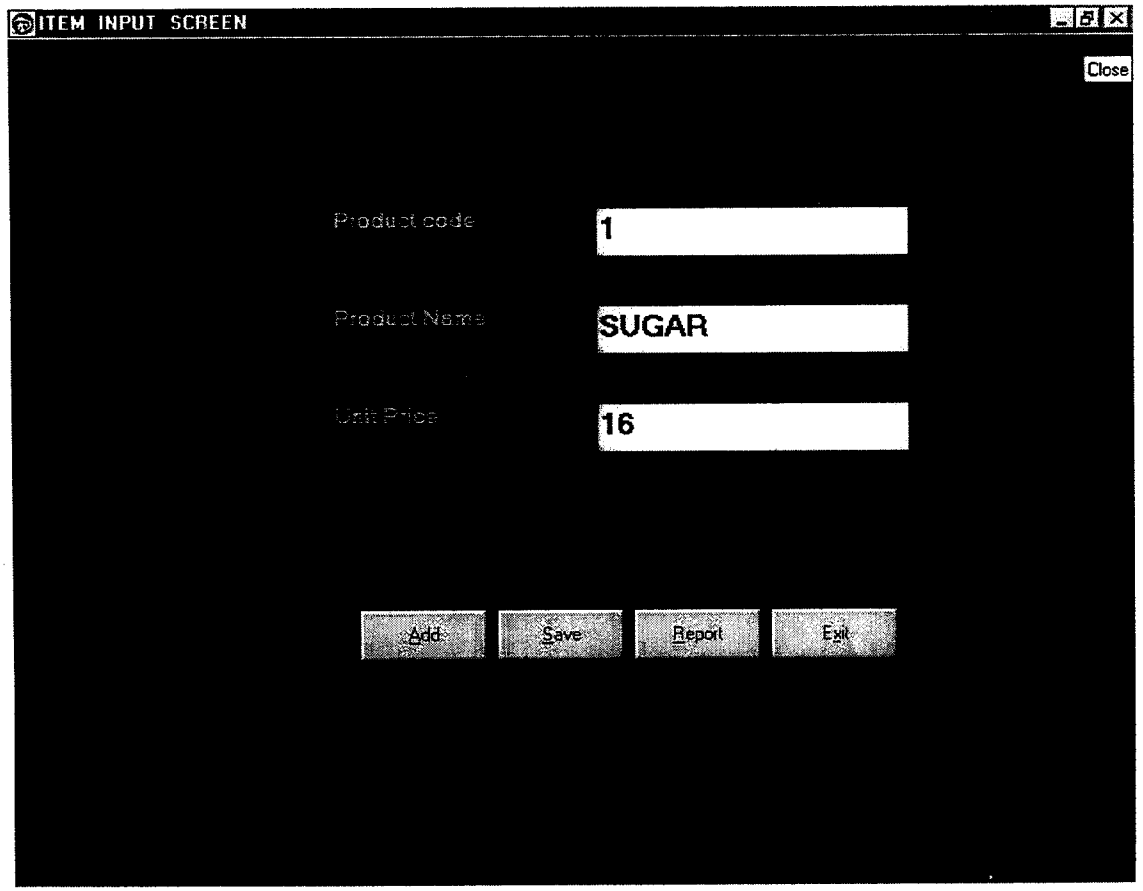
7. BIBLIOGRAPHY

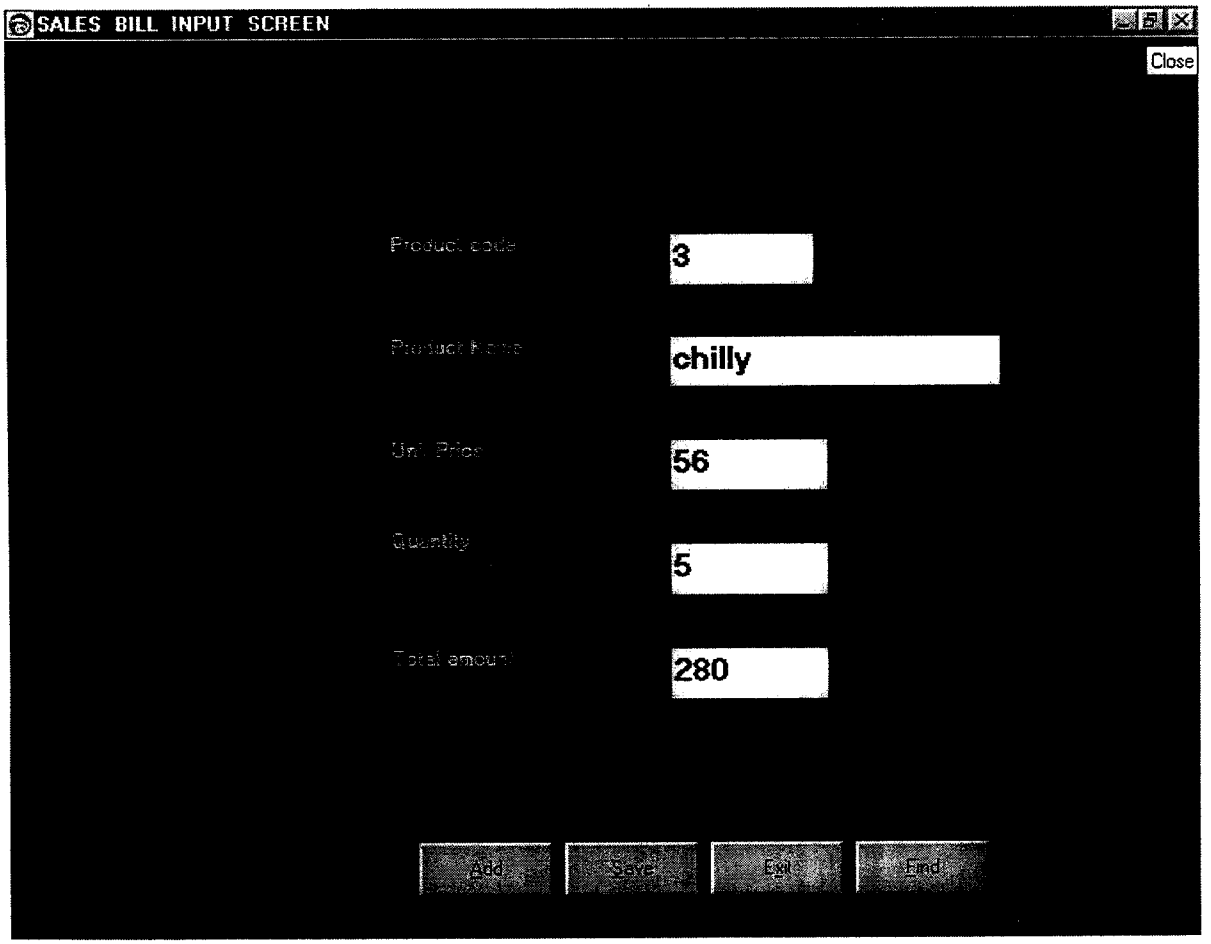
1.	BEGINNING VISUAL BASIC	PETER WRIGHTS
2.	VISUAL BASIC 6.0	GARY CORNELL
3.	MICROSOFT ACCESS	LEE AND LEE
4.	SOFTWARE ENGINEERING	ROGER S. PRESSMAN
5.	AN INTEGRATED APPROACH TO SOFTWARE ENGINEERING	PANKAJ JALOTE

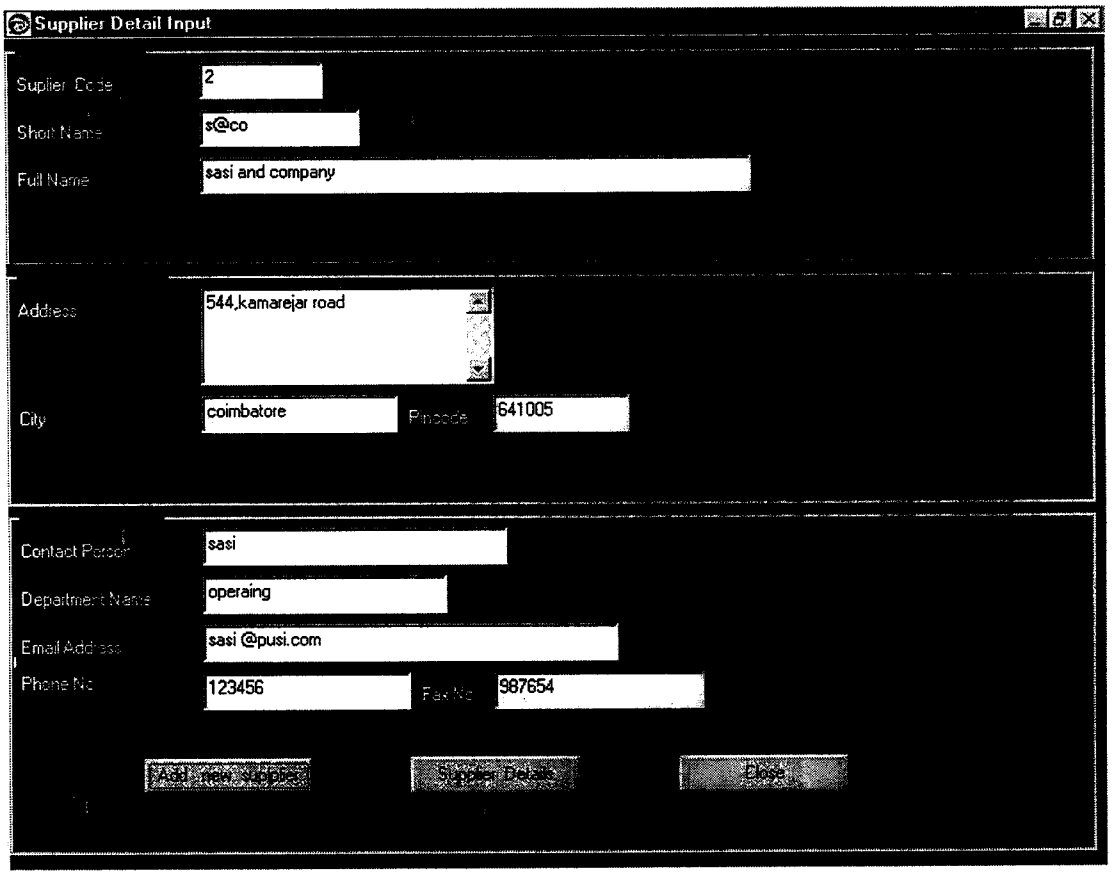
SCREENS

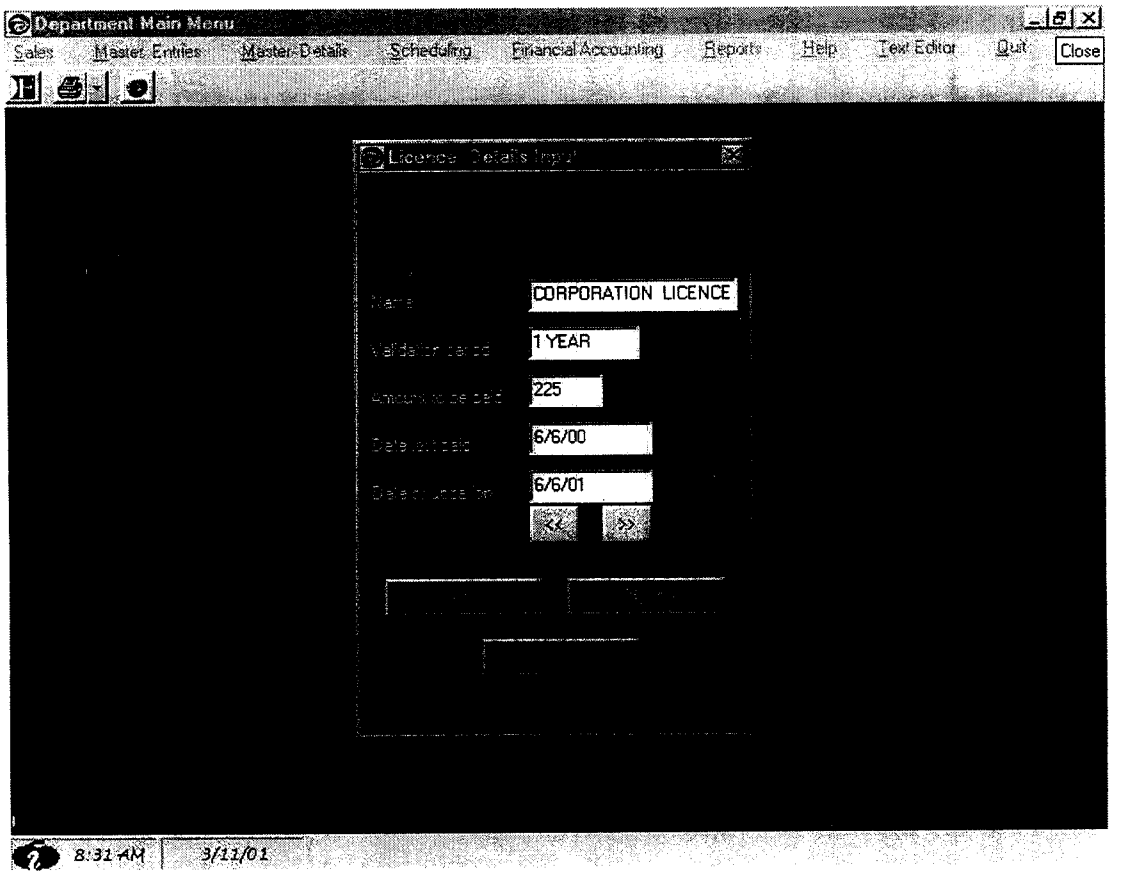
MAIN SCREEN

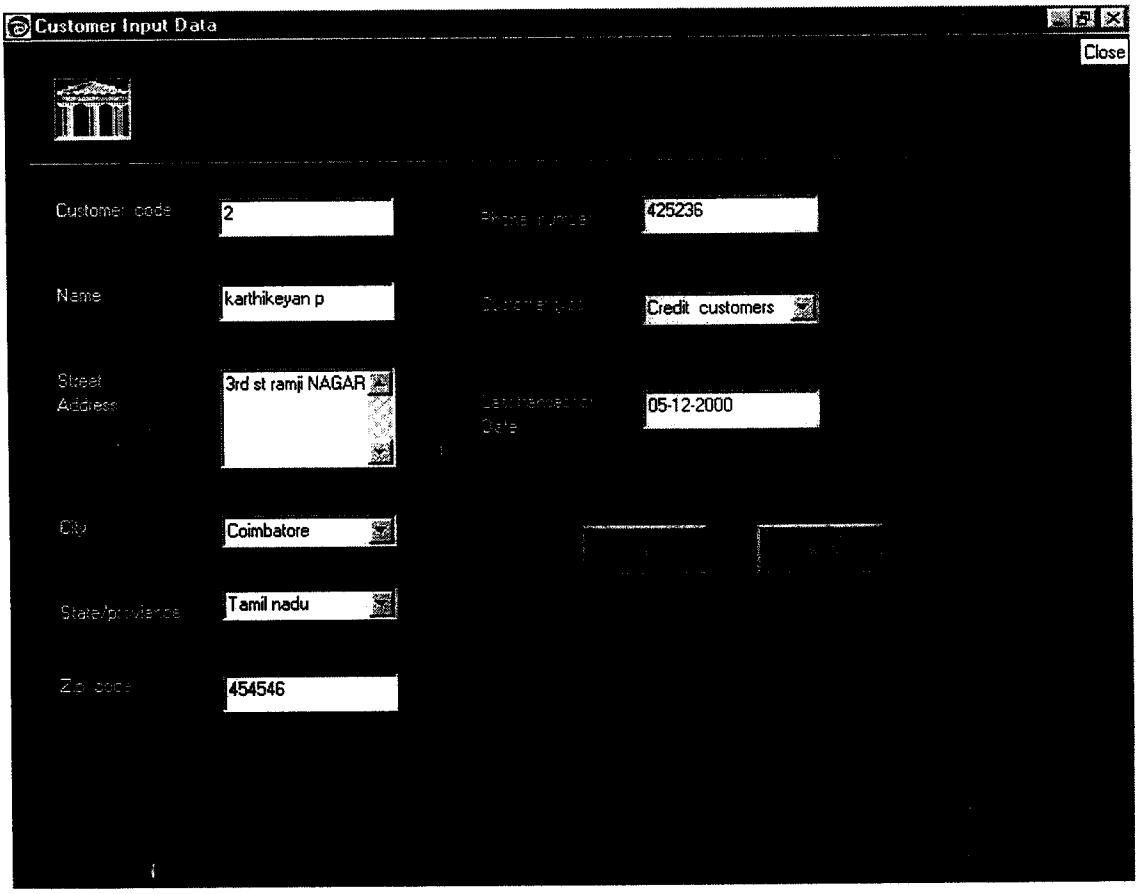

Department Main Menu
Sales
Master Entries
Master Details
Reports
Help
Quit
8:26 AM

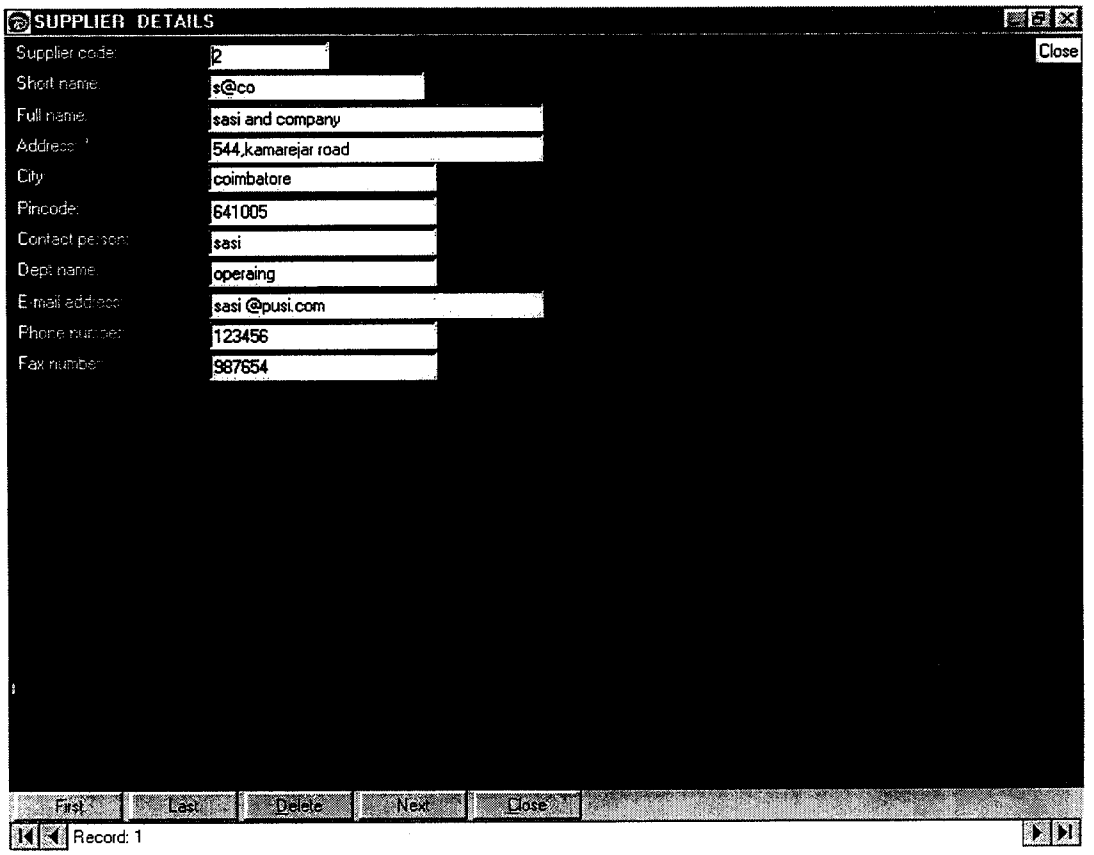

ITEM INPUT SCREEN
Close
Product code
1
Product Name
SUGAR
Unit Price
16
Add
Save
Report
Exit

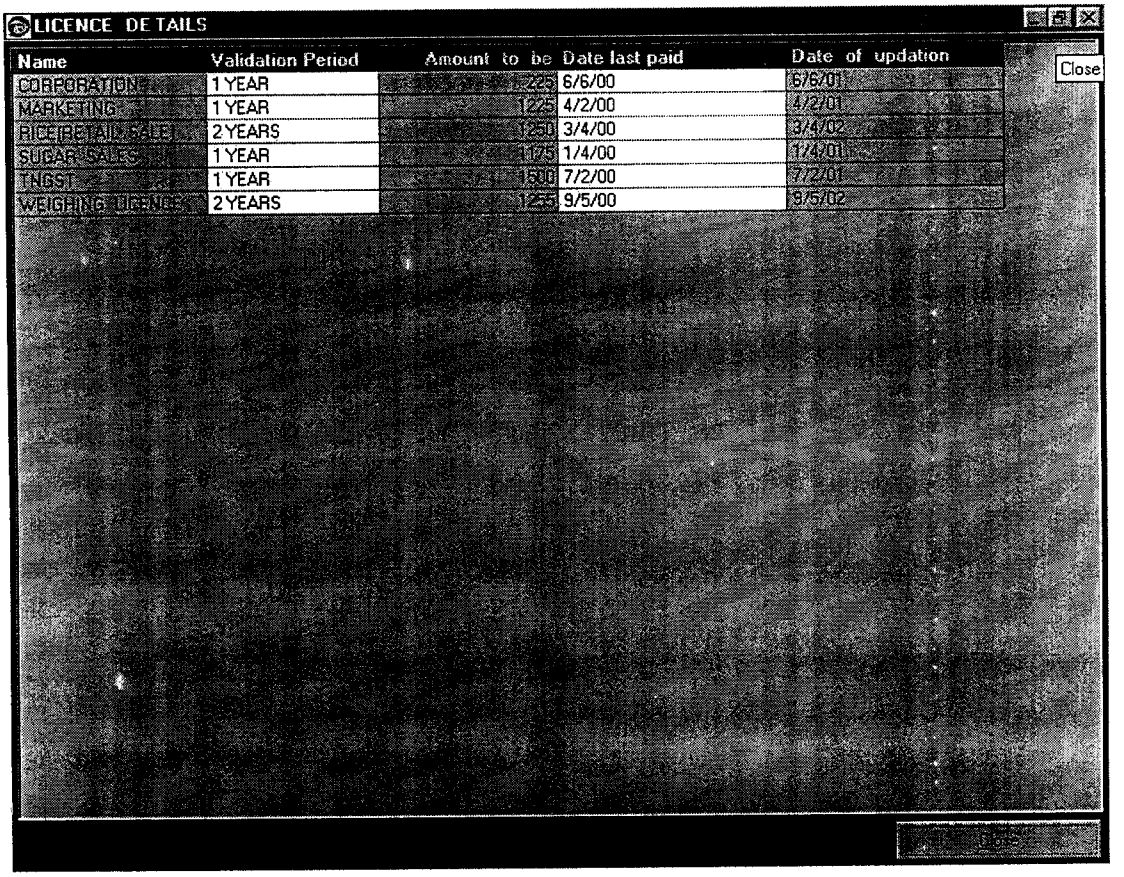

SALES BILL INPUT SCREEN
Close
Product code
3
Product Name
chilly
Unit Price
56
Quantity
5
Total amount
280
Add
Save
Exit
Find

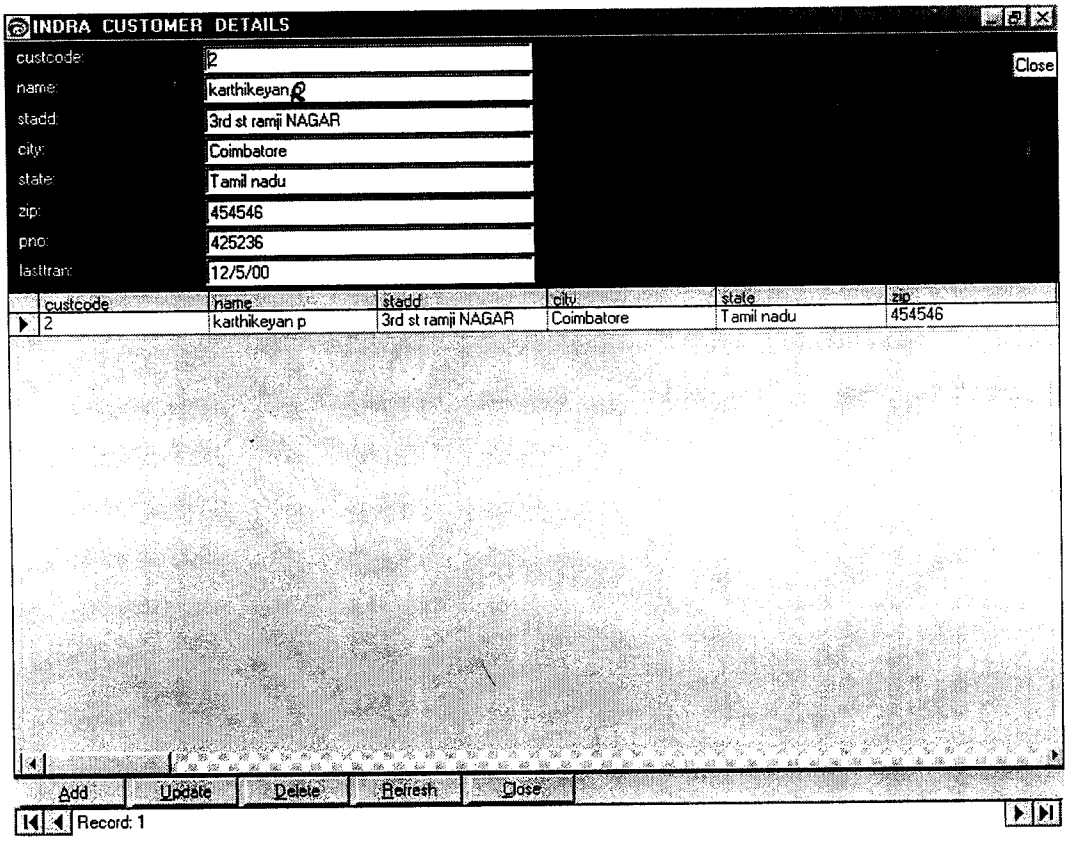

Supplier Detail Input
Suplier Code
2
Short Name
s@co
Full Name
sasi and company
Address
544,kamarejar road
City
coimbatore
Pincode
641005
Contact Person
sasi
Department Name
operaing
Email Address
sasi @pusi.com
Phone No
123456
Fax No
987654
Add New Supplier
Supplier Details
Close

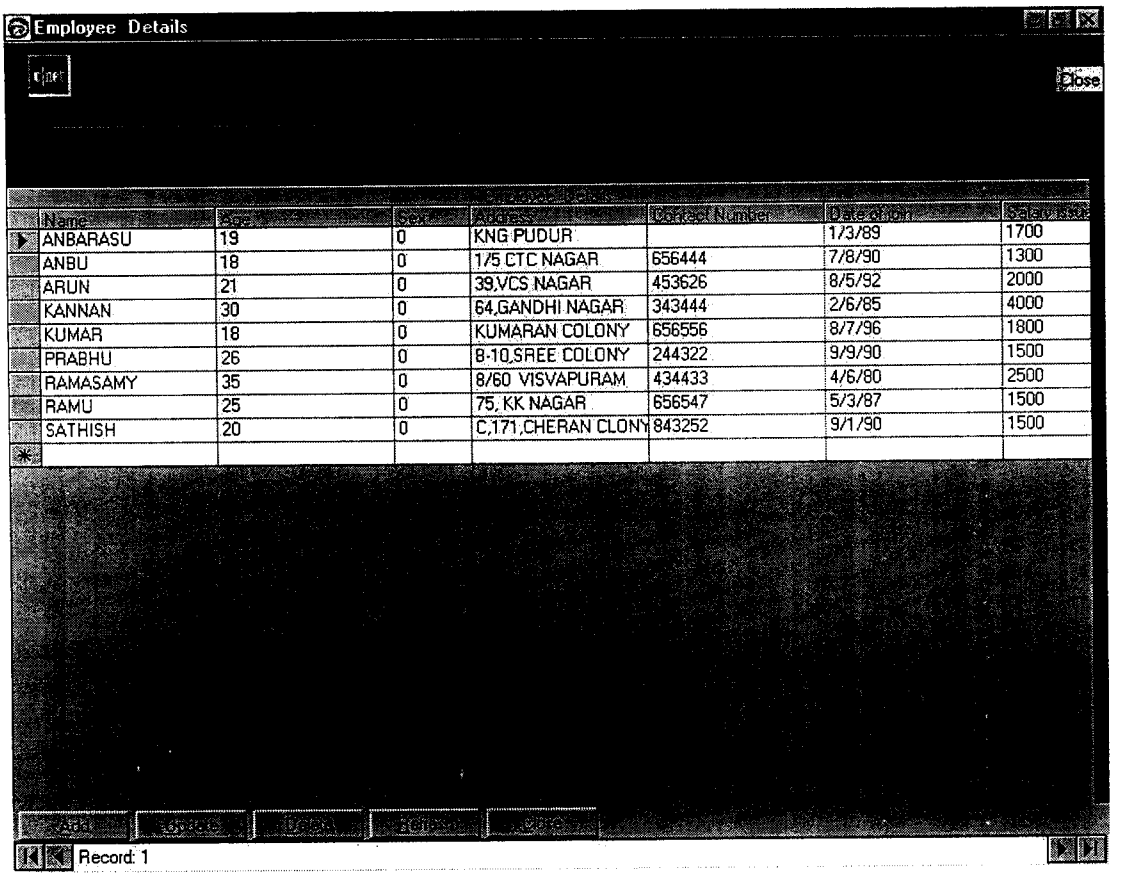

Department Main Menu
Sales
Master Entries
Master-Details
Scheduling
Financial Accounting
Reports
Help
Text Editor
Quit
Close
Licence Details Input
CORPORATION LICENCE
1 YEAR
225
6/6/00
6/6/01
8:31 AM
3/11/01

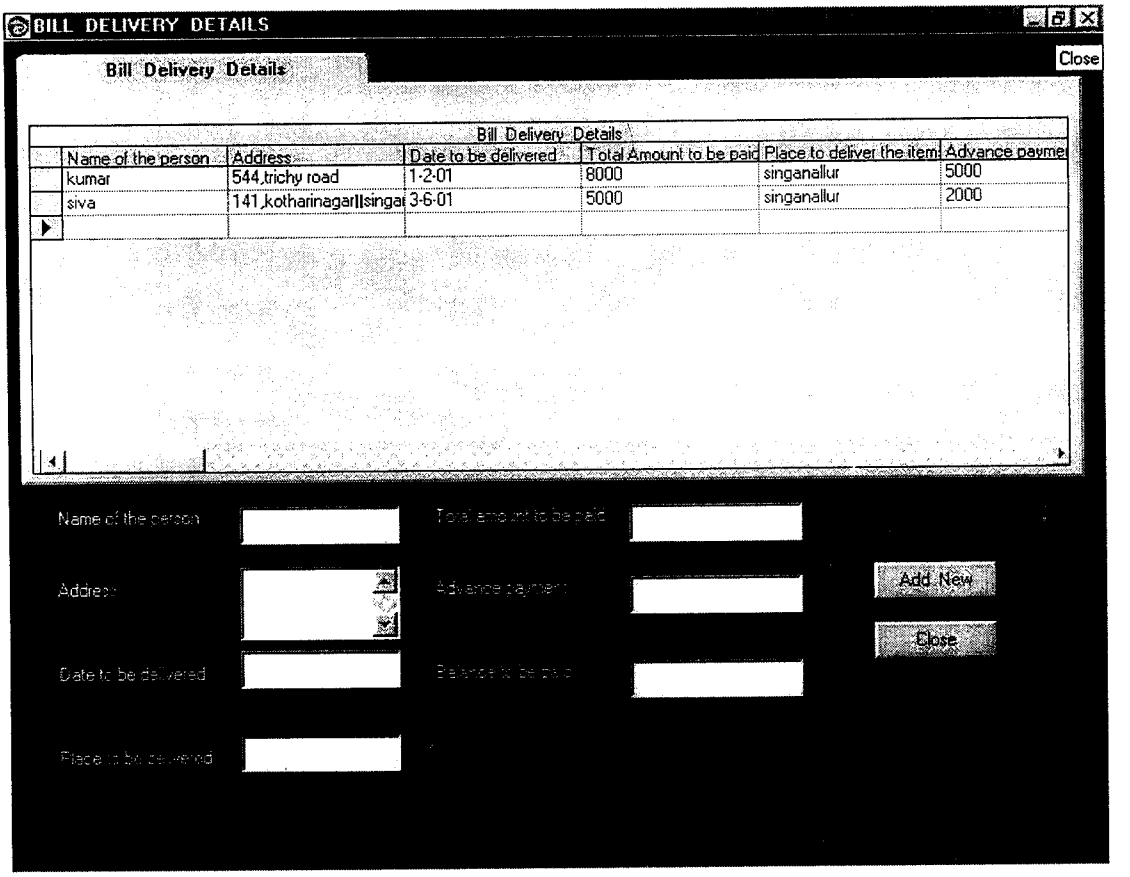

Customer Input Data
Close
Customer code
2
425236
Name
karthikeyan p
Credit customers
Street Address
3rd st ramji NAGAR
05-12-2000
City
Coimbatore
Tamil nadu
454546

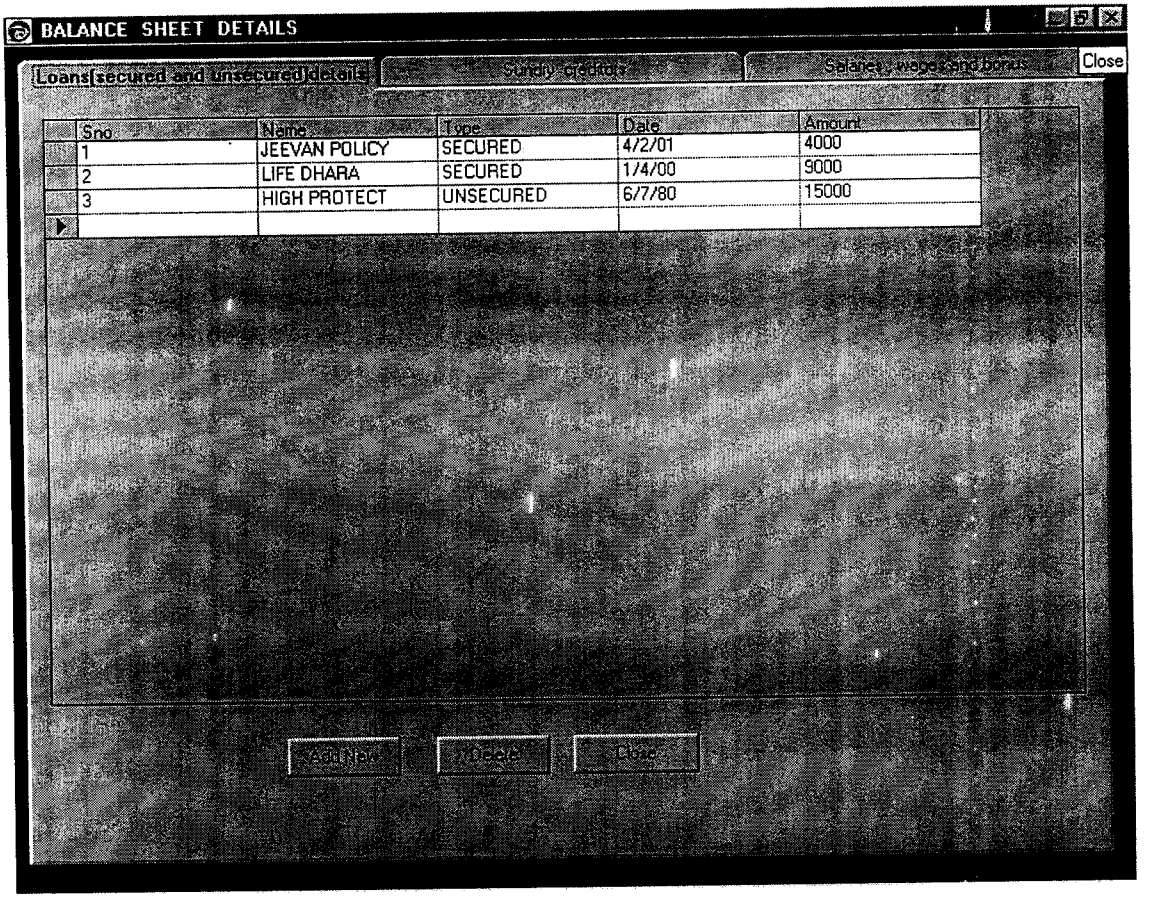

SUPPLIER DETAILS
Close
Supplier code:
2
Short name:
s@co
Full name:
sasi and company
Address:
544,kamarejar road
City:
coimbatore
Pincode:
641005
Contact person:
sasi
Dept name:
operaing
E-mail address:
sasi @pusi.com
Phone number:
123456
Fax number:
987654
First
Last
Delete
Next
Close
Record: 1

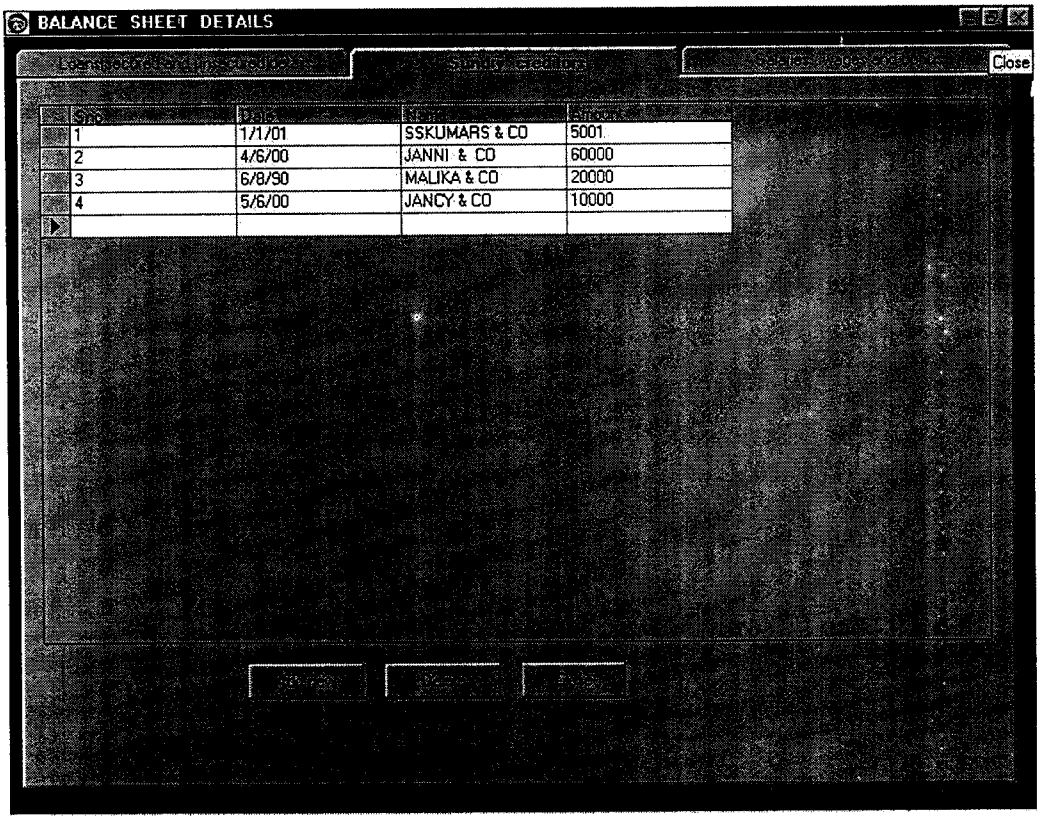

LICENCE DE TAILS
Name
Validation Period
Amount to be
Date last paid
Date of updation
CORPORATION
1 YEAR
1225
6/6/00
6/6/01
MARKETING
1 YEAR
1225
4/2/00
4/2/01
RICE(RETAIL SALE)
2 YEARS
1250
3/4/00
3/4/02
SUGAR SALES
1 YEAR
1175
1/4/00
1/4/01
TNGST
1 YEAR
1500
7/2/00
7/2/01
WEIGHING LICENCE
2 YEARS
1225
9/5/00
9/5/02
Close

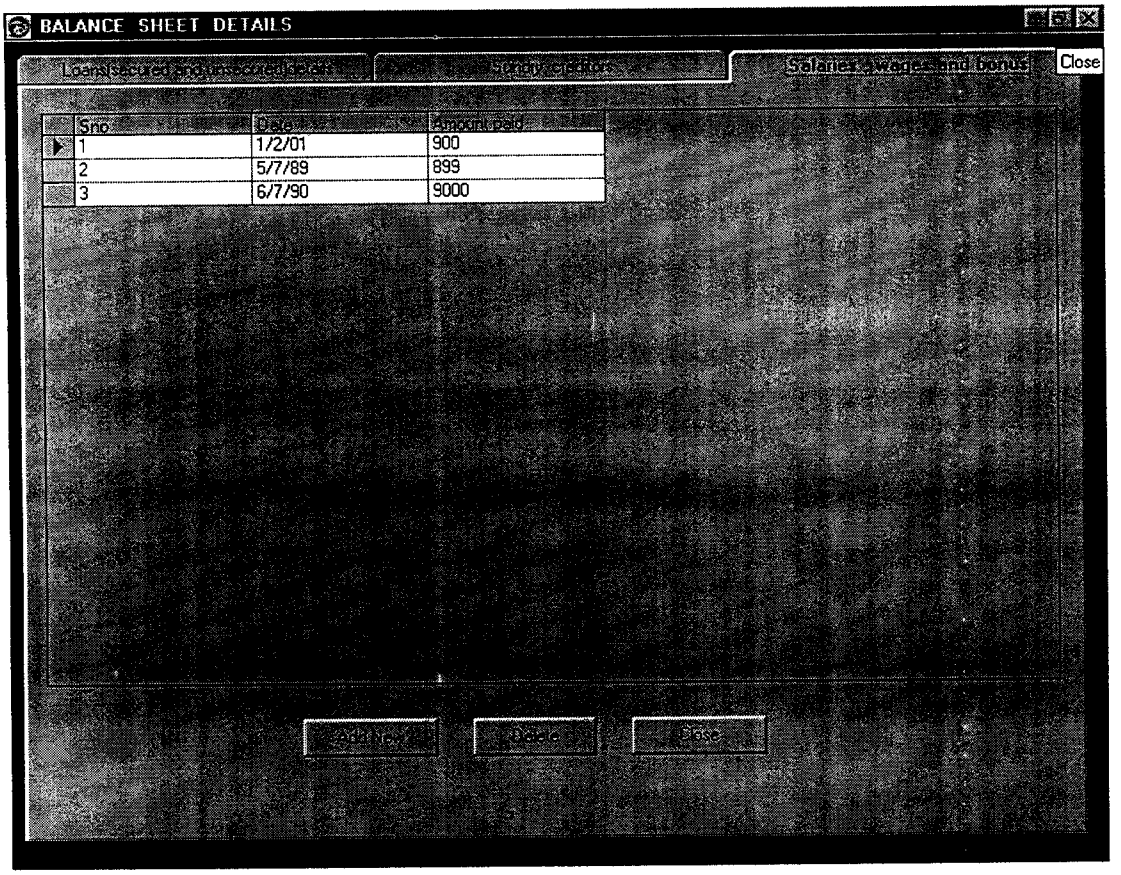

INDRA CUSTOMER DETAILS
custcode:
2
name:
karthikeyan p
stadd:
3rd st ramji NAGAR
city:
Coimbatore
state:
Tamil nadu
zip:
454546
pno:
425236
lasttran:
12/5/00
Close
custcode
name
stadd
city
state
zip
2
karthikeyan p
3rd st ramji NAGAR
Coimbatore
Tamil nadu
454546
Add
Update
Delete
Refresh
Close
Record: 1

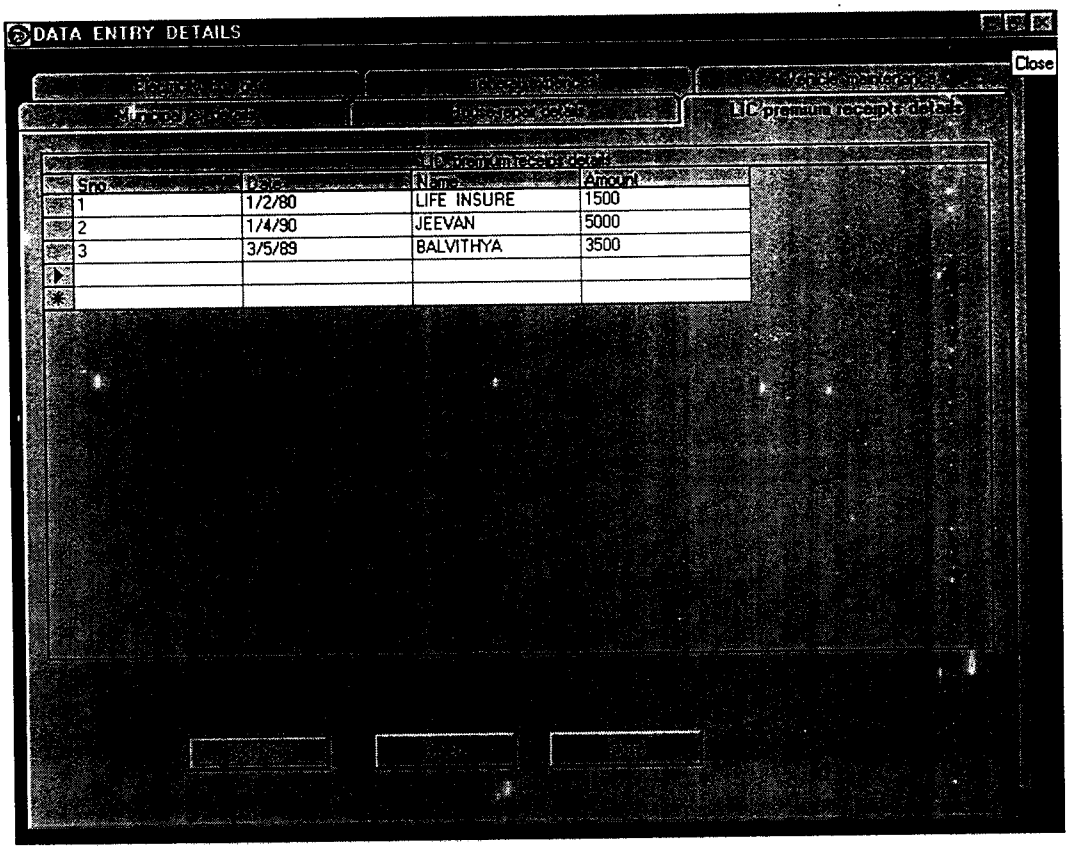

Employee Details
Close
Name
Age
Sex
Address
Contact Number
Date of Join
Salary
ANBARASU 19 0 KNG PUDUR 1/3/89 1700
ANBU 18 0 1/5 CTC NAGAR 656444 7/8/90 1300
ARUN 21 0 39,VCS NAGAR 453626 8/5/92 2000
KANNAN 30 0 64,GANDHI NAGAR 343444 2/6/85 4000
KUMAR 18 0 KUMARAN COLONY 656556 8/7/96 1800
PRABHU 26 0 8-10,SREE COLONY 244322 9/9/90 1500
RAMASAMY 35 0 8/60 VISVAPURAM 434433 4/6/80 2500
RAMU 25 0 75, KK NAGAR 656547 5/3/87 1500
SATHISH 20 0 C,171,CHERAN CLONY 843252 9/1/90 1500
Add
Update
Delete
Refresh
Save
Record: 1

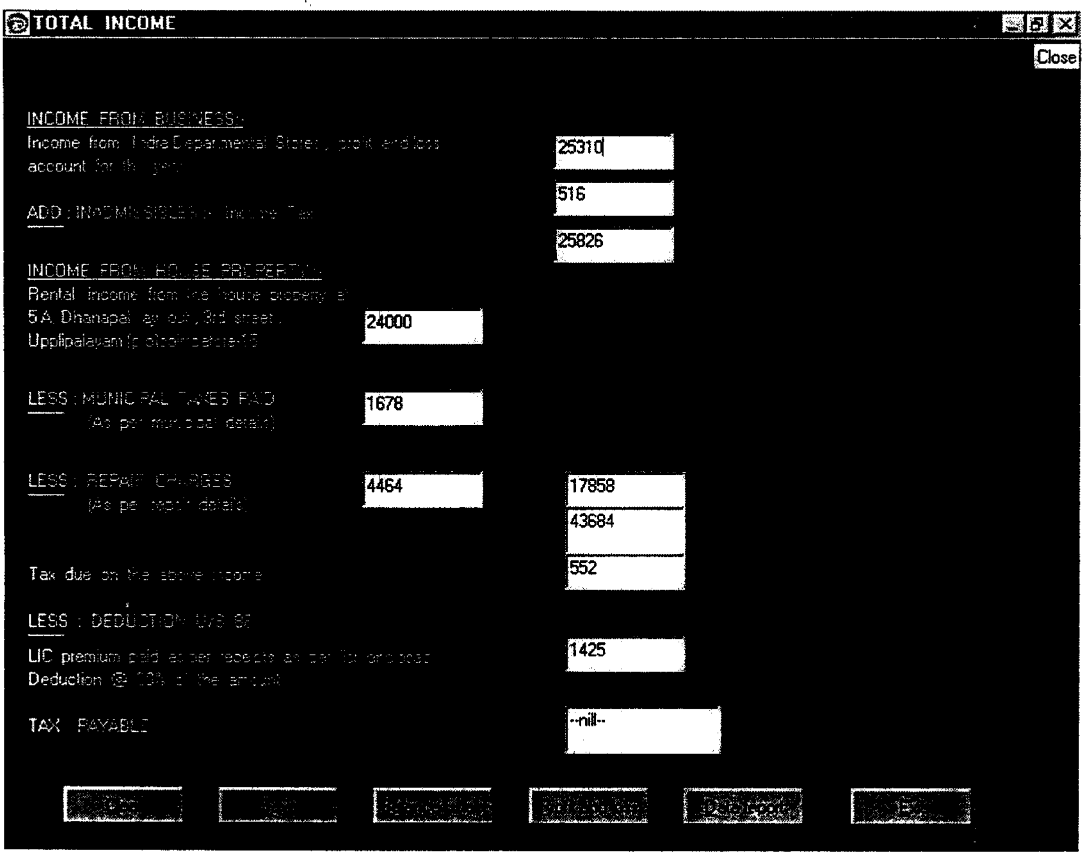

BILL DELIVERY DETAILS
Close
Bill Delivery Details
Bill Delivery Details
Name of the person
Address
Date to be delivered
Total Amount to be paid
Place to deliver the item
Advance payme
kumar
544,trichy road
1-2-01
8000
singanallur
5000
siva
141,kotharinagar||singa
3-6-01
5000
singanallur
2000
Name of the person
Address
Date to be delivered
Place to be delivered
Total amount to be paid
Advance payment
Balance to be paid
Add New
Close


BALANCE SHEET DETAILS
Loans(secured and unsecured)details
Sundry creditors
Salaries,wages and bonus
Close
Sno
Name
Type
Date
Amount
1
JEEVAN POLICY
SECURED
4/2/01
4000
2
LIFE DHARA
SECURED
1/4/00
9000
3
HIGH PROTECT
UNSECURED
6/7/80
15000
Add New
Delete
Close

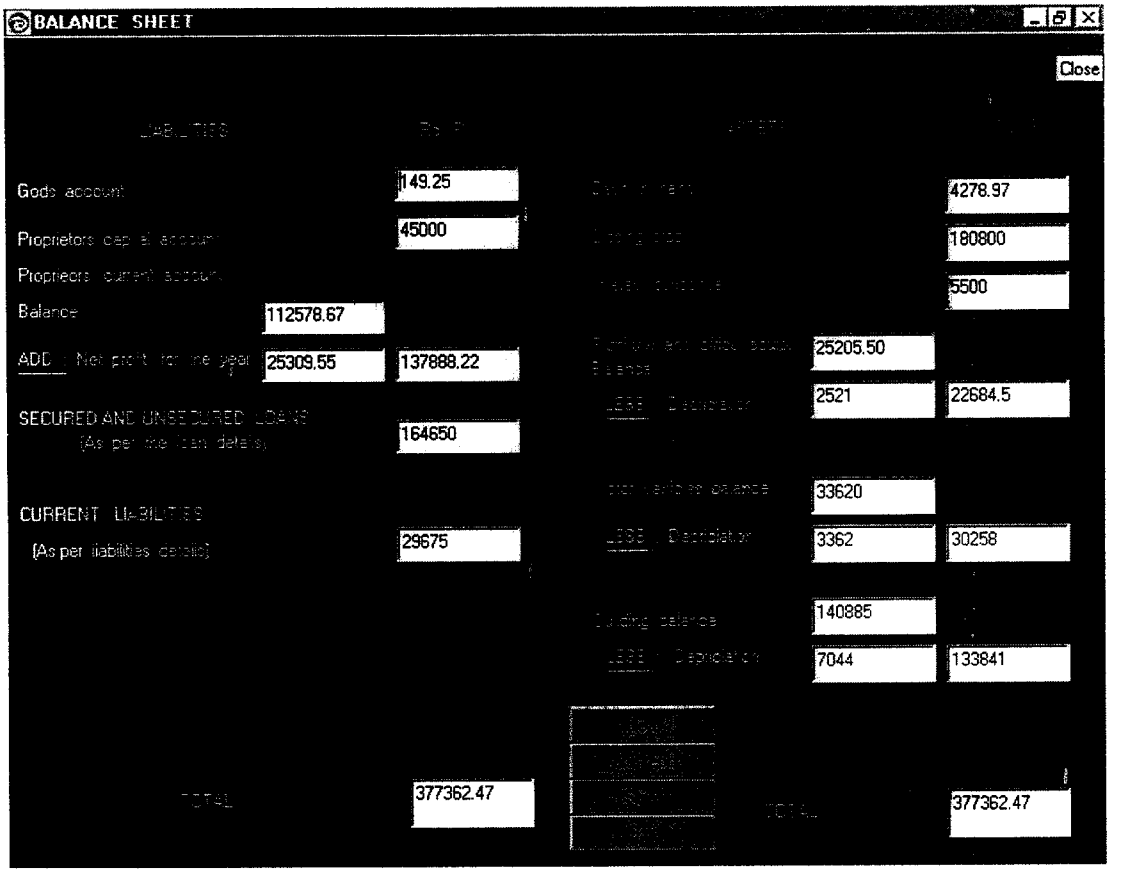

BALANCE SHEET DETAILS
Close
1
1/1/01
SSKUMARS & CO
5001
2
4/6/00
JANNI & CO
60000
3
6/8/90
MALIKA & CO
20000
4
5/6/00
JANCY & CO
10000

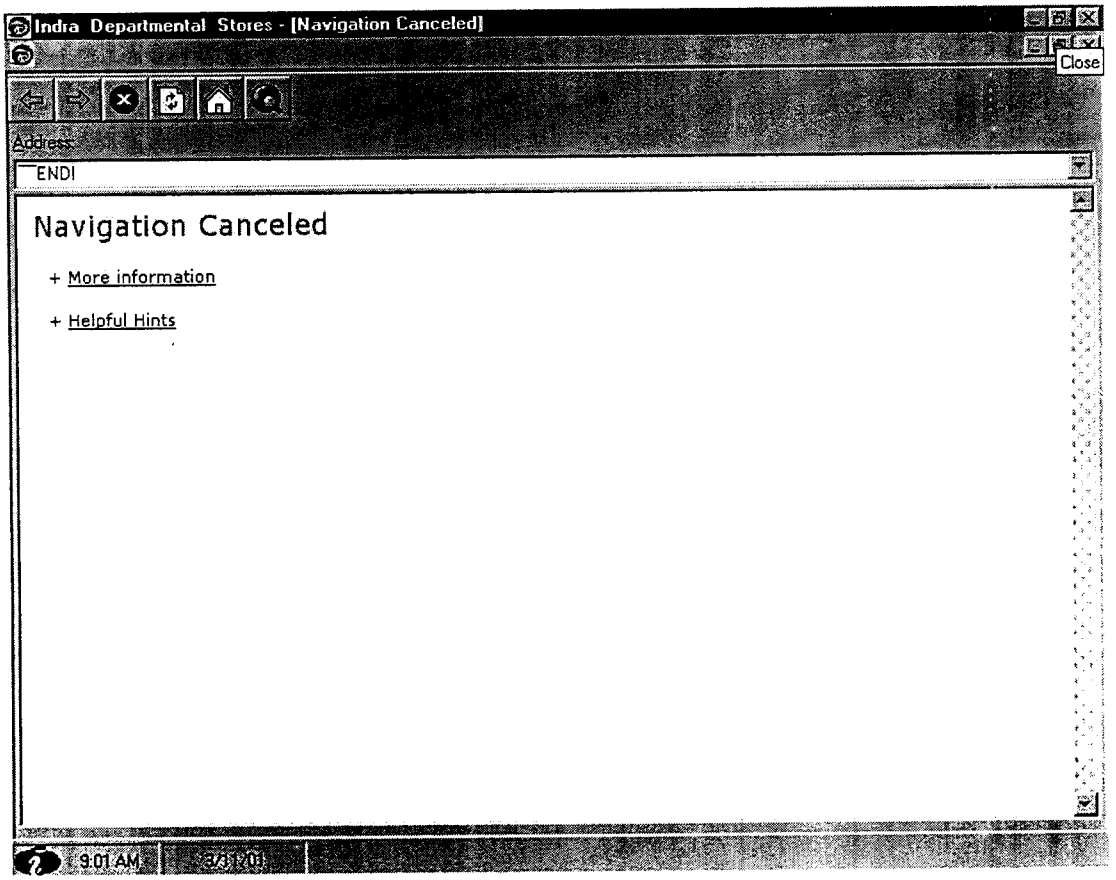

BALANCE SHEET DETAILS
Salaries,wages and bonus
Close
Sno
Date
Amount paid
1
1/2/01
900
2
5/7/89
899
3
6/7/90
9000
Add new
Delete
Close


DATA ENTRY DETAILS
Close
LIC premium receipts details
Sno
Date
Name
Amount
1
1/2/80
LIFE INSURE
1500
2
1/4/90
JEEVAN
5000
3
3/5/89
BALVITHYA
3500



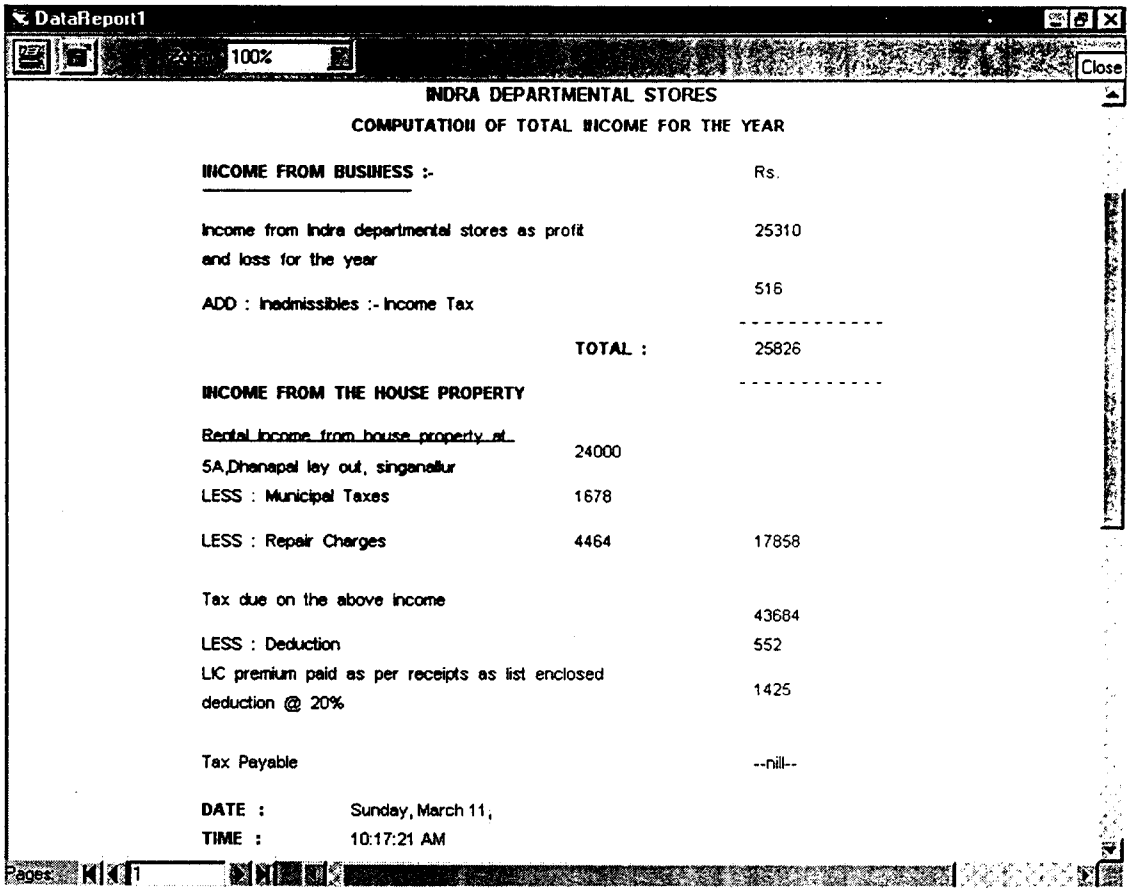
TOTAL INCOME
Close
INCOME FROM BUSINESS:-
25310
516
25826
INCOME FROM HOUSE PROPERTY:-
Rental income from the house property at
5A Dhanapal lay out, 3rd street,
24000
LESS : MUNICIPAL TAXES PAID
1678
LESS : REPAIR CHARGES
4464
17858
43684
Tax due on the above income
552
LESS : DEDUCTION U/S 88
1425
Deduction @ 10% of the amount
TAX PAYABLE
--nill--


BALANCE SHEET
Close
LIABILITIES
Rs.P
ASSETS
Gods account
149.25
4278.97
Proprietors cap al account
45000
180800
Proprietors current account
Balance
112578.67
5500
ADD Net profit for the year
25309.55
137888.22
25205.50
LESS Drawings
2521
22684.5
SECURED AND UNSECURED LOANS
(As per the loan details)
164650
33620
CURRENT LIABILITES
(As per liabilities details)
29675
LESS Depriciation
3362
30258
140885
LESS Depriciation
7044
133841
TOTAL
377362.47
TOTAL
377362.47

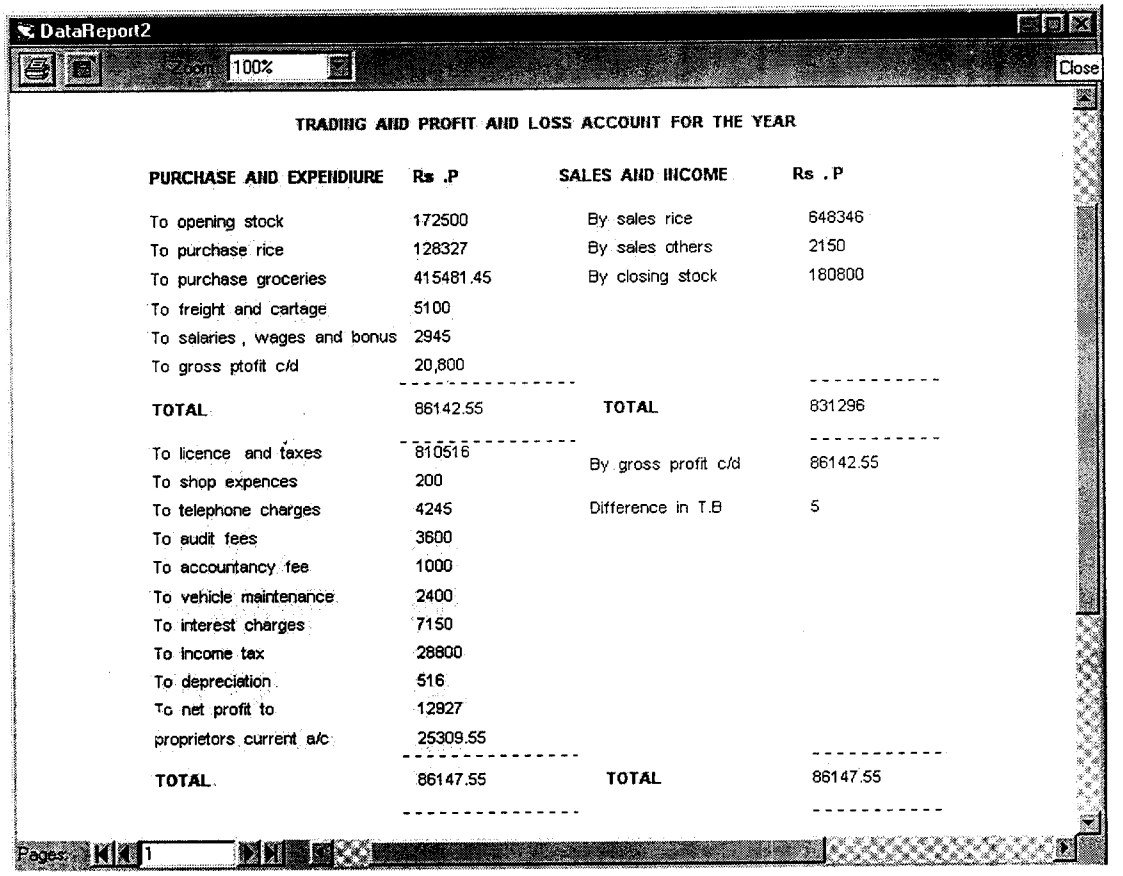

Indra Departmental Stores - [Navigation Canceled]
Close
Address
ENDI
Navigation Canceled
+ More information
+ Helpful Hints
9:01 AM

REPORTS

COMPUTATION OF TOTAL INCOME



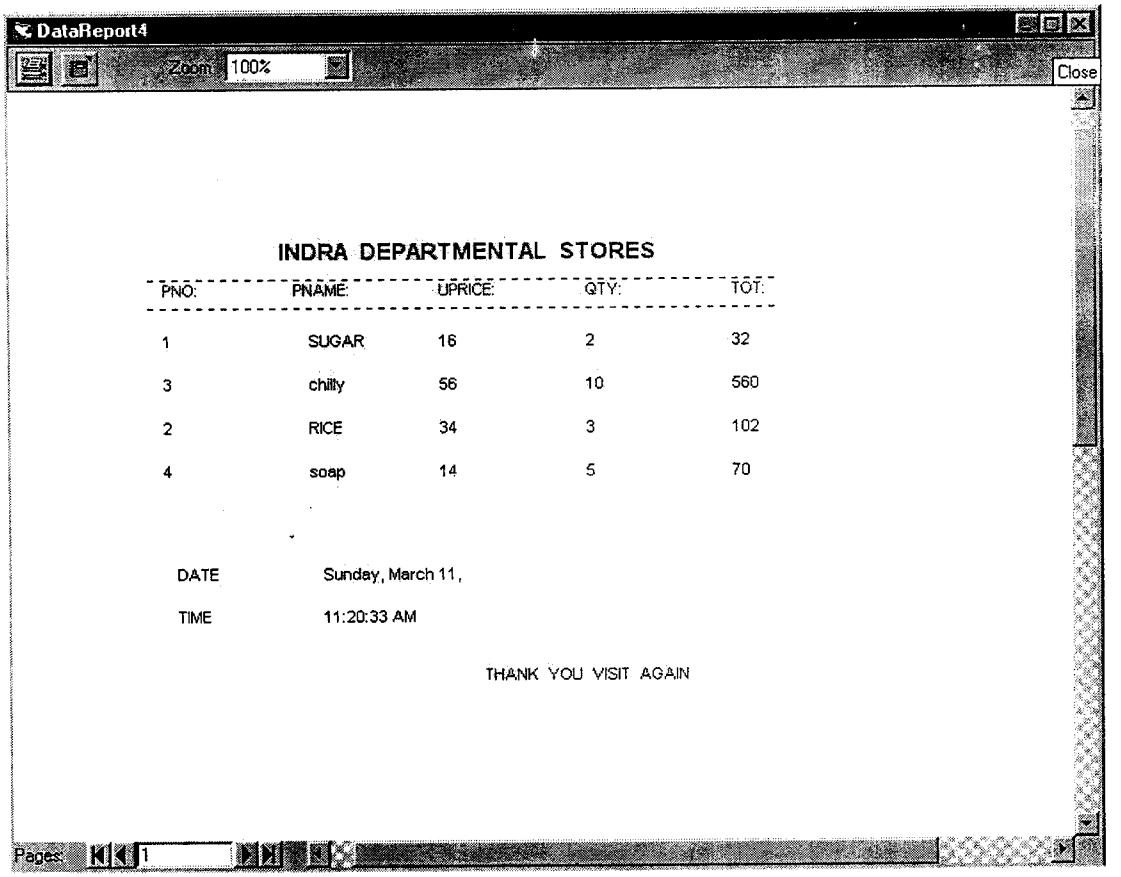
TRADING, PROFIT AND LOSS ACCOUNT



TRADING AND PROFIT AND LOSS ACCOUNT FOR THE YEAR

PURCHASE AND EXPENDIURE	Rs .P	SALES AND INCOME	Rs . P
To opening stock	172500	By sales rice	648346
To purchase rice	128327	By sales others	2150
To purchase groceries	415481.45	By closing stock	180800
To freight and cartage	5100		
To salaries , wages and bonus	2945		
To gross ptofit c/d	20,800		
TOTAL	86142.55	**TOTAL**	831296
To licence and taxes	810516	By gross profit c/d	86142.55
To shop expences	200		
To telephone charges	4245	Difference in T.B	5
To audit fees	3600		
To accountancy fee	1000		
To vehicle maintenance	2400		
To interest charges	7150		
To income tax	28800		
To depreciation	516		
To net profit to	12927		
proprietors current a/c	25309.55		
TOTAL	86147.55	**TOTAL**	86147.55

GENERATION OF SALES BILL



SOURCE CODE

```
Private Sub employee1_Click()
frmemployee.Show
End Sub

Private Sub item_Click()
Form5.Show
End Sub

Private Sub item1_Click()
frmitem.Show
End Sub

Private Sub licence_Click()
Form4.Show
End Sub

Private Sub licence1_Click()
frmlicense.Show
End Sub

Private Sub Quit1_Click()
End
End Sub

Private Sub sched_Click()
Form14.Show
End Sub

Private Sub supplier_Click()
Form7.Show
End Sub

Private Sub supplier1_Click()
frmsupplier.Show
End Sub

Private Sub texted_Click()
Dim tmp As String
tmp = Shell("C:\My Documents\kumars\notepad.exe", vbMaximizedFocus)
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComCtlLib.Button)
   On Error Resume Next
   Select Case Button.Key
      'Case "Find"
```

```
Public tm As String

Private Sub abcb_Click()
Form15.Show
End Sub

Private Sub agdre_Click()
Form16.Show
End Sub

Private Sub balance1_Click()
DataReport3.Show
End Sub

Private Sub balsheet_Click()
Form9.Show
End Sub

Private Sub bill_Click()
Form13.Show
End Sub

Private Sub calculator_Click()
Dim tmp As String
tmp = Shell("C:\WINDOWS\CALC.EXE", vbMaximizedFocus)
End Sub

Private Sub customer_Click()
form2.Show
End Sub

Private Sub customer1_Click()
frmindracust.Show
End Sub

Private Sub dentry_Click()
Form8.Show
End Sub

Private Sub eadsf_Click()
DataReport4.Show
End Sub

Private Sub employee_Click()
Form6.Show
End Sub
```

```
      Case "fort"
        Form10.Show
      'Case "Print"

      Case "dscvy"
        frmBrowser.Show

   End Select
End Sub


'Private Sub cascade_Click()
'Dim tmp As String
'tmp = Shell("C:\WINDOWS\CALC.EXE", vbMaximizedFocus)
'End Sub


Private Sub contents_Click()
form2.Show
End Sub

Private Sub new_Click()
Form4.Show
End Sub


Private Sub open_Click()
Form6.Show
End Sub

Private Sub search_Click()
frmindracust.Show
End Sub

Private Sub about_Click()
Form3.Show
End Sub

Private Sub exit_Click()
End
End Sub
Private Sub Form_MouseMove(Button As Integer, shift As Integer, X As Single, Y
As Single)
 Label14.FontSize = 14
End Sub
```

```
Private Sub Timer1_Timer()
Label15.Left = Label15.Left - 50
If Label15.Left < 0 Then
Label15.Left = 9000
Label15.Left = Label15.Left - 50
End If
End Sub

Private Sub tip_Click()
frmTip.Show
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
   Dim nRet As Integer
   'if there is no helpfile for this project display a message to the user
   'you can set the HelpFile for your application in the
   'Project Properties dialog
   If Len(App.HelpFile) = 0 Then
     MsgBox "Unable to display Help Contents. There is no Help associated with
this project.", vbInformation, Me.Caption
   Else
      On Error Resume Next
      nRet = OSWinHelp(Me.hWnd, App.HelpFile, 261, 0)
      If Err Then
         MsgBox Err.Description
      End If
   End If
End Sub

Private Sub mnuFileExit_Click()
   'unload the form
   Unload Me
End Sub
Private Sub mnuFilePrint_Click()
   On Error Resume Next
   If ActiveForm Is Nothing Then Exit Sub
      With dlgCommonDialog
      .DialogTitle = "Print"
      .CancelError = True
      .Flags = cdlPDReturnDC + cdlPDNoPageNums
      If ActiveForm.rtfText.SelLength = 0 Then
         .Flags = .Flags + cdlPDAllPages
      Else
         .Flags = .Flags + cdlPDSelection
      End If
      .ShowPrinter
```

```
    If Err <> MSComDlg.cdlCancel Then
      ActiveForm.rtfText.SelPrint .hDC
    End If
  End With

End Sub


Private Sub mnuFilePageSetup_Click()
  On Error Resume Next
  With dlgCommonDialog
    .DialogTitle = "Page Setup"
    .CancelError = True
    .ShowPrinter
  End With

End Sub



Private Sub mnuFileSaveAs_Click()
  Dim sFile As String
    If ActiveForm Is Nothing Then Exit Sub
    With dlgCommonDialog
    .DialogTitle = "Save As"
    .CancelError = False
    'ToDo: set the flags and attributes of the common dialog control
    .Filter = "All Files (*.*)|*.*"
    .ShowSave
    If Len(.FileName) = 0 Then
      Exit Sub
    End If
    sFile = .FileName
  End With
  ActiveForm.Caption = sFile
  ActiveForm.rtfText.SaveFile sFile

End Sub

Private Sub mnuFileSave_Click()
  Dim sFile As String
  If Left$(ActiveForm.Caption, 8) = "Document" Then
    With dlgCommonDialog
      .DialogTitle = "Save"
      .CancelError = False
      'ToDo: set the flags and attributes of the common dialog control
```

```
        .Filter = "All Files (*.*)|*.*"
        .ShowSave
        If Len(.FileName) = 0 Then
           Exit Sub
        End If
        sFile = .FileName
      End With
      ActiveForm.rtfText.SaveFile sFile
   Else
      sFile = ActiveForm.Caption
      ActiveForm.rtfText.SaveFile sFile
   End If

End Sub

Private Sub mnuFileClose_Click()
   'ToDo: Add 'mnuFileClose_Click' code.
   MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
   'Dim sFile As String
   'If ActiveForm Is Nothing Then LoadNewDoc
    '  With dlgCommonDialog
    '  .DialogTitle = "Open"
    '  .CancelError = False
      'ToDo: set the flags and attributes of the common dialog control
     ' .Filter = "All Files (*.*)|*.*"
    '  .ShowOpen
     ' If Len(.FileName) = 0 Then
     '    Exit Sub
    '  End If
    '  sFile = .FileName
   'End With
   'ActiveForm.rtfText.LoadFile sFile
   'ActiveForm.Caption = sFile

End Sub

Private Sub mnuFileNew_Click()
  ' LoadNewDoc
End Sub

Private Sub toolbar_Click()
Form5.Show
End Sub
```

```
        .Filter = "All Files (*.*)|*.*"
        .ShowSave
        If Len(.FileName) = 0 Then
           Exit Sub
        End If
        sFile = .FileName
      End With
      ActiveForm.rtfText.SaveFile sFile
   Else
      sFile = ActiveForm.Caption
      ActiveForm.rtfText.SaveFile sFile
   End If

End Sub

Private Sub mnuFileClose_Click()
   'ToDo: Add 'mnuFileClose_Click' code.
   MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
   'Dim sFile As String
   'If ActiveForm Is Nothing Then LoadNewDoc
    '  With dlgCommonDialog
    '  .DialogTitle = "Open"
    '  .CancelError = False
      'ToDo: set the flags and attributes of the common dialog control
    '  .Filter = "All Files (*.*)|*.*"
    '  .ShowOpen
    '  If Len(.FileName) = 0 Then
    '     Exit Sub
    '  End If
    '  sFile = .FileName
   'End With
   'ActiveForm.rtfText.LoadFile sFile
   'ActiveForm.Caption = sFile

End Sub

Private Sub mnuFileNew_Click()
  ' LoadNewDoc
End Sub

Private Sub toolbar_Click()
Form5.Show
End Sub
```

```
Private Sub undo_Click()
Form7.Show
End Sub

Private Sub webbrowser_Click()
frmBrowser.Show
End Sub

Private Sub totincome_Click()
DataReport1.Show
End Sub

Private Sub trading_Click()
DataReport2.Show
End Sub


Public tm As String

Private Sub abcb_Click()
Form15.Show
End Sub

Private Sub agdre_Click()
Form16.Show
End Sub

Private Sub balance1_Click()
DataReport3.Show
End Sub

Private Sub balsheet_Click()
Form9.Show
End Sub

Private Sub bill_Click()
Form13.Show
End Sub

Private Sub calculator_Click()
Dim tmp As String
tmp = Shell("C:\WINDOWS\CALC.EXE", vbMaximizedFocus)
End Sub
```

```
Private Sub customer_Click()
form2.Show
End Sub

Private Sub customer1_Click()
frmindracust.Show
End Sub

Private Sub dentry_Click()
Form8.Show
End Sub

Private Sub eadsf_Click()
DataReport4.Show
End Sub

Private Sub employee_Click()
Form6.Show
End Sub

Private Sub employee1_Click()
frmemployee.Show
End Sub

Private Sub item_Click()
Form5.Show
End Sub

Private Sub item1_Click()
frmitem.Show
End Sub

Private Sub licence_Click()
Form4.Show
End Sub

Private Sub licence1_Click()
frmlicense.Show
End Sub

Private Sub Quit1_Click()
End
End Sub
```

```
Private Sub sched_Click()
Form14.Show
End Sub

Private Sub supplier_Click()
Form7.Show
End Sub

Private Sub supplier1_Click()
frmsupplier.Show
End Sub

Private Sub texted_Click()
Dim tmp As String
tmp = Shell("C:\My Documents\kumars\notepad.exe", vbMaximizedFocus)
End Sub

Private Sub Toolbar1_ButtonClick(ByVal Button As MSComCtlLib.Button)
   On Error Resume Next
   Select Case Button.Key
      'Case "Find"

      Case "fort"
        Form10.Show
      'Case "Print"

      Case "dscvy"
        frmBrowser.Show

   End Select
End Sub


'Private Sub cascade_Click()
'Dim tmp As String
'tmp = Shell("C:\WINDOWS\CALC.EXE", vbMaximizedFocus)
'End Sub


Private Sub contents_Click()
form2.Show
End Sub

Private Sub new_Click()
Form4.Show
End Sub
```

```
Private Sub open_Click()
Form6.Show
End Sub

Private Sub search_Click()
frmindracust.Show
End Sub

Private Sub about_Click()
Form3.Show
End Sub

Private Sub exit_Click()
End
End Sub
Private Sub Form_MouseMove(Button As Integer, shift As Integer, X As Single, Y
As Single)
 Label14.FontSize = 14
End Sub
Private Sub Label10_MouseMove(Button As Integer, shift As Integer, X As Single,
Y As Single)
'Label10.ForeColor = RGB(255, 0, 0)
End Sub
Private Sub Label11_Click()
'form2.Show
End Sub
Private Sub Label11_MouseMove(Button As Integer, shift As Integer, X As Single,
Y As Single)
'Label11.ForeColor = RGB(255, 0, 0)
End Sub
Private Sub Label12_MouseMove(Button As Integer, shift As Integer, X As Single,
Y As Single)
'Label12.ForeColor = RGB(255, 0, 0)
End Sub
Private Sub Label13_MouseMove(Button As Integer, shift As Integer, X As Single,
Y As Single)
'Label13.ForeColor = RGB(255, 0, 0)
End Sub
Private Sub Label14_Click()
End
End Sub
Private Sub Label14_MouseMove(Button As Integer, shift As Integer, X As Single,
Y As Single)
Label14.FontSize = 18
End Sub
```

```
Private Sub Label16_MouseMove(Button As Integer, shift As Integer, X As Single,
Y As Single)
'Label16.ForeColor = RGB(255, 0, 0)
End Sub
Private Sub Label9_MouseMove(Button As Integer, shift As Integer, X As Single, Y
As Single)
'Label9.ForeColor = RGB(255, 0, 0)
End Sub
Private Sub shedule_Click()
'Form4.Show
End Sub


Private Sub tilehori_Click()
'Dim tm As String
tm = Shell("C:\Program Files\WINAMP\winamp.exe", vbMaximizedFocus)
End Sub

'Private Sub tilevert_Click()
'Dim tmp As String
'tmp = Shell("C:\My Documents\kumars\notepad.exe", vbMaximizedFocus)
'End Sub

Private Sub Timer1_Timer()
'Label15.Left = Label15.Left - 50
'If Label15.Left < 0 Then
'Label15.Left = 9000
'Label15.Left = Label15.Left - 50
'End If
End Sub

Private Sub tip_Click()
frmTip.Show
End Sub

Private Sub mnuHelpSearchForHelpOn_Click()
  ' Dim nRet As Integer


   'if there is no helpfile for this project display a message to the user
   'you can set the HelpFile for your application in the
   'Project Properties dialog
   'If Len(App.HelpFile) = 0 Then
     ' MsgBox "Unable to display Help Contents. There is no Help associated with
this project.", vbInformation, Me.Caption
  ' Else
```

```
      'On Error Resume Next
      'nRet = OSWinHelp(Me.hWnd, App.HelpFile, 261, 0)
     ' If Err Then
       '   MsgBox Err.Description
      'End If
  '  End If

End Sub



Private Sub mnuFileExit_Click()
   'unload the form
   Unload Me

End Sub



Private Sub mnuFilePrint_Click()
   On Error Resume Next
   If ActiveForm Is Nothing Then Exit Sub


   With dlgCommonDialog
      .DialogTitle = "Print"
      .CancelError = True
      .Flags = cdlPDReturnDC + cdlPDNoPageNums
      If ActiveForm.rtfText.SelLength = 0 Then
         .Flags = .Flags + cdlPDAllPages
      Else
         .Flags = .Flags + cdlPDSelection
      End If
      .ShowPrinter
      If Err <> MSComDlg.cdlCancel Then
         ActiveForm.rtfText.SelPrint .hDC
      End If
   End With

End Sub


Private Sub mnuFilePageSetup_Click()
   On Error Resume Next
   With dlgCommonDialog
```

```
    .DialogTitle = "Page Setup"
    .CancelError = True
    .ShowPrinter
  End With

End Sub



Private Sub mnuFileSaveAs_Click()
  Dim sFile As String
    If ActiveForm Is Nothing Then Exit Sub
    With dlgCommonDialog
    .DialogTitle = "Save As"
    .CancelError = False
    'ToDo: set the flags and attributes of the common dialog control
    .Filter = "All Files (*.*)|*.*"
    .ShowSave
    If Len(.FileName) = 0 Then
      Exit Sub
    End If
    sFile = .FileName
  End With
  ActiveForm.Caption = sFile
  ActiveForm.rtfText.SaveFile sFile

End Sub

Private Sub mnuFileSave_Click()
  Dim sFile As String
  If Left$(ActiveForm.Caption, 8) = "Document" Then
    With dlgCommonDialog
      .DialogTitle = "Save"
      .CancelError = False
      'ToDo: set the flags and attributes of the common dialog control
      .Filter = "All Files (*.*)|*.*"
      .ShowSave
      If Len(.FileName) = 0 Then
        Exit Sub
      End If
      sFile = .FileName
    End With
    ActiveForm.rtfText.SaveFile sFile
  Else
    sFile = ActiveForm.Caption
    ActiveForm.rtfText.SaveFile sFile
```

```
    End If

End Sub

Private Sub mnuFileClose_Click()
    'ToDo: Add 'mnuFileClose_Click' code.
    MsgBox "Add 'mnuFileClose_Click' code."
End Sub

Private Sub mnuFileOpen_Click()
    'Dim sFile As String
    'If ActiveForm Is Nothing Then LoadNewDoc
     '  With dlgCommonDialog
     '  .DialogTitle = "Open"
     '  .CancelError = False
        'ToDo: set the flags and attributes of the common dialog control
     '  .Filter = "All Files (*.*)|*.*"
     '  .ShowOpen
     '  If Len(.FileName) = 0 Then
     '     Exit Sub
     '  End If
     '  sFile = .FileName
    'End With
    'ActiveForm.rtfText.LoadFile sFile
    'ActiveForm.Caption = sFile

End Sub

Private Sub mnuFileNew_Click()
   ' LoadNewDoc
End Sub

Private Sub toolbar_Click()
Form5.Show
End Sub

Private Sub undo_Click()
Form7.Show
End Sub

Private Sub webbrowser_Click()
frmBrowser.Show
End Sub

Private Sub totincome_Click()
DataReport1.Show
```

```
End Sub

Private Sub Text8_GotFocus()
Dim d, e, f, g
d = Val(Text1.Text) + Val(Text2.Text)
e = Val(Text3.Text) + Val(Text4.Text)
f = Val(Text5.Text) + Val(Text6.Text)
g = d + e + f
Text8.Text = g + Val(Text7.Text)
End Sub



Dim cn As New ADODB.Connection
Dim rs As New ADODB.Recordset
Private Sub Command1_Click()
Text1.SetFocus
End Sub

Private Sub Command2_Click()
rs.Fields(0) = Text1.Text
rs.Fields(1) = Text2.Text
rs.Fields(2) = Text3.Text
m = MsgBox("DO YOU WANT TO ADD THIS RECORD?", vbYesNo, "SAVE")
If m = vbYes Then
rs.Update
rs.AddNew
MsgBox " YOUR RECORD IS ADDED"
Call clr
Else
rs.CancelUpdate
MsgBox " YOUR RECORD IS NOT ADDED"
Call clr
End If
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Command4_Click()
rs.Close
Text1.Enabled = False
a = InputBox("Enter the Product Code", "EDIT")
rs.Open "select * from salmast where pno='" & a & "'", cn, adOpenDynamic,
adLockOptimistic, adCmdText
```

```
rs1.Fields(1) = Text2.Text
rs1.Fields(2) = Text3.Text
rs1.Fields(3) = Text4.Text
rs1.Fields(4) = Text5.Text
m = MsgBox("DO YOU WANT TO ADD THIS RECORD?", vbYesNo, "SAVE")
If m = vbYes Then
rs1.Update
rs1.AddNew
MsgBox " YOUR RECORD IS ADDED"
Call clr
Else
rs1.CancelUpdate
MsgBox " YOUR RECORD IS NOT ADDED"
Call clr
End If
End Sub

Private Sub Command3_Click()
End
End Sub

Private Sub Command4_Click()
DataReport4.Show
End Sub

Private Sub Form_Load()
With cn
.CursorLocation = adUseClient
.ConnectionTimeout = 30
.Provider = "MSDAORA"
.Open "", "scott", "tiger"
End With
rs.Open "select * from salmast", cn, adOpenKeyset, adLockOptimistic, adCmdText
With cn1
.CursorLocation = adUseClient
.ConnectionTimeout = 30
.Provider = "MSDAORA"
.Open "", "scott", "tiger"
End With
rs1.Open "select * from sale", cn1, adOpenKeyset, adLockOptimistic, adCmdText
End Sub

Private Sub Text5_GotFocus()
Text5.Text = Val(Text3.Text) * Val(Text4.Text)
End Sub
```

```
Public Sub clr()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
End Sub

Private Sub Command1_Click()
Adodc1.Recordset.AddNew
Text1.SetFocus
End Sub

Private Sub Command2_Click()
p = MsgBox("ARE YOU SURE TO DELETE  THE  RECORD?", vbOKCancel,
Delete)
If p = vbOK Then
Adodc1.Recordset.Delete
End If
End Sub

Private Sub Command3_Click()
Me.Hide
End Sub

Private Sub Command4_Click()
If Not Adodc1.Recordset.EOF Then
Adodc1.Recordset.MoveNext
End If
End Sub


Private Sub Command1_Click()
Adodc1.Recordset.AddNew
'text1.SetFocus
End Sub
Private Sub clr()
Text1.Text = ""
Text2.Text = ""
Text3.Text = ""
Text4.Text = ""
Text5.Text = ""
Text6.Text = ""
Text7.Text = ""
Text8.Text = ""
End Sub
```

```
Private Sub Command2_Click()
frmemployee.Show
End Sub

Private Sub Command3_Click()
Me.Hide
End Sub
Private Sub Command1_Click()
Adodc1.Recordset.AddNew
text1.SetFocus
End Sub

Private Sub Command2_Click()
frmsupplier.Show
End Sub

Private Sub Command3_Click()
Me.Hide
End Sub

Private Sub Form_Load()
'text1.SetFocus
End Sub


Private Sub Command1_Click()
Adodc5.Recordset.AddNew
End Sub

Private Sub Command16_Click()
Adodc3.Recordset.AddNew
End Sub

Private Sub Command17_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
Adodc3.Recordset.Delete
End If
End Sub

Private Sub Command18_Click()
Me.Hide
End Sub
```

```
Private Sub Command2_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
Adodc5.Recordset.Delete
End If
End Sub

Private Sub Command6_Click()
Me.Hide
End Sub

Private Sub Command7_Click()
Adodc6.Recordset.AddNew
End Sub

Private Sub Command8_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
Adodc6.Recordset.Delete
End If
End Sub

Private Sub Command9_Click()
Me.Hide
End Sub
Private Sub Command3_Click()
Adodc4.Recordset.AddNew
End Sub

Private Sub Command4_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
Adodc4.Recordset.Delete
End If
End Sub

Private Sub Command5_Click()
Me.Hide
End Sub
Private Sub Command13_Click()
Adodc2.Recordset.AddNew
End Sub

Private Sub Command14_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
```

```
Adodc2.Recordset.Delete
End If
End Sub

Private Sub Command15_Click()
Me.Hide
End Sub

Private Sub Command10_Click()
Adodc1.Recordset.AddNew
End Sub

Private Sub Command11_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
Adodc1.Recordset.Delete
End If
End Sub

Private Sub Command12_Click()
Me.Hide
End Sub


Private Sub Command1_Click()
Adodc1.Recordset.AddNew
End Sub

Private Sub Command2_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
Adodc1.Recordset.Delete
End If
End Sub

Private Sub Command6_Click()
Me.Hide
End Sub

Private Sub Command3_Click()
Adodc2.Recordset.AddNew
End Sub

Private Sub Command4_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
```

```
Adodc2.Recordset.Delete
End If
End Sub

Private Sub Command5_Click()
Me.Hide
End Sub
Private Sub Command7_Click()
Adodc3.Recordset.AddNew
End Sub

Private Sub Command8_Click()
p = MsgBox("Are u sure to delete?", vbOKCancel, Indra - stores)
If p = vbOK Then
Adodc3.Recordset.Delete
End If
End Sub

Private Sub Command9_Click()
Me.Hide
End Sub

Private Sub DataGrid1_Click()

End Sub

Option Explicit
Public StartingAddress As String
Dim mbDontNavigateNow As Boolean
Private Sub Form_Load()
   On Error Resume Next
   Me.Show
   tbToolBar.Refresh
   Form_Resize
   cboAddress.Move 50, lblAddress.Top + lblAddress.Height + 15
   If Len(StartingAddress) > 0 Then
      cboAddress.Text = StartingAddress
      cboAddress.AddItem cboAddress.Text
      'try to navigate to the starting address
      timTimer.Enabled = True
      brwWebBrowser.Navigate StartingAddress
   End If
End Sub
Private Sub brwWebBrowser_DownloadComplete()
   On Error Resume Next
   Me.Caption = brwWebBrowser.LocationName
```

```
End Sub
Private Sub brwWebBrowser_NavigateComplete(ByVal URL As String)
  Dim i As Integer
  Dim bFound As Boolean
  Me.Caption = brwWebBrowser.LocationName
  For i = 0 To cboAddress.ListCount - 1
    If cboAddress.List(i) = brwWebBrowser.LocationURL Then
      bFound = True
      Exit For
    End If
  Next i
  mbDontNavigateNow = True
  If bFound Then
    cboAddress.RemoveItem i
  End If
  cboAddress.AddItem brwWebBrowser.LocationURL, 0
  cboAddress.ListIndex = 0
  mbDontNavigateNow = False
End Sub
Private Sub cboAddress_Click()
  If mbDontNavigateNow Then Exit Sub
  timTimer.Enabled = True
  brwWebBrowser.Navigate cboAddress.Text
End Sub
Private Sub cboAddress_KeyPress(KeyAscii As Integer)
  On Error Resume Next
  If KeyAscii = vbKeyReturn Then
    cboAddress_Click
  End If
End Sub
Private Sub Form_Resize()
  cboAddress.Width = Me.ScaleWidth - 100
  brwWebBrowser.Width = Me.ScaleWidth - 100
  brwWebBrowser.Height = Me.ScaleHeight - (picAddress.Top +
picAddress.Height) - 100
End Sub
Private Sub timTimer_Timer()
  If brwWebBrowser.Busy = False Then
    timTimer.Enabled = False
    Me.Caption = brwWebBrowser.LocationName
  Else
    Me.Caption = "Working..."
  End If
End Sub
Private Sub tbToolBar_ButtonClick(ByVal Button As Button)
  On Error Resume Next
```

```
  timTimer.Enabled = True

  Select Case Button.Key
    Case "Back"
      brwWebBrowser.GoBack
    Case "Forward"
      brwWebBrowser.GoForward
    Case "Refresh"
      brwWebBrowser.Refresh
    Case "Home"
      brwWebBrowser.GoHome
    Case "Search"
      brwWebBrowser.GoSearch
    Case "Stop"
      timTimer.Enabled = False
      brwWebBrowser.Stop
      Me.Caption = brwWebBrowser.LocationName
  End Select
End Sub


Private Sub Form_Resize()
 On Error Resume Next
 'This will resize the grid when the form is resized
 grdDataGrid.Height = Me.ScaleHeight - datPrimaryRS.Height - 30 -
picButtons.Height
End Sub

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

Private Sub datPrimaryRS_Error(ByVal ErrorNumber As Long, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
ByVal HelpContext As Long, fCancelDisplay As Boolean)
 '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:" & Description
End Sub

Private Sub datPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This will display the current record position for this recordset
```

```
 datPrimaryRS.Caption = "Record: " &
CStr(datPrimaryRS.Recordset.AbsolutePosition)
End Sub

Private Sub datPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This is where you put validation code
 'This event gets called when the following actions occur
 Dim bCancel As Boolean

 Select Case adReason
 Case adRsnAddNew
 Case adRsnClose
 Case adRsnDelete
 Case adRsnFirstChange
 Case adRsnMove
 Case adRsnRequery
 Case adRsnResynch
 Case adRsnUndoAddNew
 Case adRsnUndoDelete
 Case adRsnUndoUpdate
 Case adRsnUpdate
 End Select

 If bCancel Then adStatus = adStatusCancel
End Sub

Private Sub cmdAdd_Click()
 On Error GoTo AddErr
 datPrimaryRS.Recordset.MoveLast
 grdDataGrid.SetFocus
 SendKeys "{down}"

 Exit Sub
AddErr:
 MsgBox Err.Description
End Sub

Private Sub cmdDelete_Click()
 On Error GoTo DeleteErr
 With datPrimaryRS.Recordset
  .Delete
  .MoveNext
  If .EOF Then .MoveLast
 End With
```

```
  Exit Sub
DeleteErr:
  MsgBox Err.Description
End Sub

Private Sub cmdRefresh_Click()
  'This is only needed for multi user apps
  On Error GoTo RefreshErr
  datPrimaryRS.Refresh
  Exit Sub
RefreshErr:
  MsgBox Err.Description
End Sub

Private Sub cmdUpdate_Click()
  On Error GoTo UpdateErr

  datPrimaryRS.Recordset.UpdateBatch adAffectAll
  Exit Sub
UpdateErr:
  MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
  Unload Me
End Sub


Private Sub Form_Load()
  Set grdDataGrid.DataSource =
datPrimaryRS.Recordset("ChildCMD").UnderlyingValue
End Sub

Private Sub Form_Resize()
  On Error Resume Next
  'This will resize the grid when the form is resized
  grdDataGrid.Width = Me.ScaleWidth
  grdDataGrid.Height = Me.ScaleHeight - grdDataGrid.Top - datPrimaryRS.Height
- 30 - picButtons.Height
End Sub

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

```
Private Sub datPrimaryRS_Error(ByVal ErrorNumber As Long, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
ByVal HelpContext As Long, fCancelDisplay As Boolean)
  '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:" & Description
End Sub

Private Sub datPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  'This will display the current record position for this recordset
  datPrimaryRS.Caption = "Record: " &
CStr(datPrimaryRS.Recordset.AbsolutePosition)
End Sub

Private Sub datPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
  'This is where you put validation code
  'This event gets called when the following actions occur
  Dim bCancel As Boolean

  Select Case adReason
  Case adRsnAddNew
  Case adRsnClose
  Case adRsnDelete
  Case adRsnFirstChange
  Case adRsnMove
  Case adRsnRequery
  Case adRsnResynch
  Case adRsnUndoAddNew
  Case adRsnUndoDelete
  Case adRsnUndoUpdate
  Case adRsnUpdate
  End Select

  If bCancel Then adStatus = adStatusCancel
End Sub

Private Sub cmdAdd_Click()
  On Error GoTo AddErr
  datPrimaryRS.Recordset.AddNew

  Exit Sub
```

```
Private Sub datPrimaryRS_Error(ByVal ErrorNumber As Long, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
ByVal HelpContext As Long, fCancelDisplay As Boolean)
 '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:" & Description
End Sub

Private Sub datPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This will display the current record position for this recordset
 datPrimaryRS.Caption = "Record: " &
CStr(datPrimaryRS.Recordset.AbsolutePosition)
End Sub

Private Sub datPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This is where you put validation code
 'This event gets called when the following actions occur
 Dim bCancel As Boolean

 Select Case adReason
 Case adRsnAddNew
 Case adRsnClose
 Case adRsnDelete
 Case adRsnFirstChange
 Case adRsnMove
 Case adRsnRequery
 Case adRsnResynch
 Case adRsnUndoAddNew
 Case adRsnUndoDelete
 Case adRsnUndoUpdate
 Case adRsnUpdate
 End Select

 If bCancel Then adStatus = adStatusCancel
End Sub

Private Sub cmdAdd_Click()
 On Error GoTo AddErr
 datPrimaryRS.Recordset.AddNew

 Exit Sub
```

```
AddErr:
 MsgBox Err.Description
End Sub

Private Sub cmdDelete_Click()
 On Error GoTo DeleteErr
 With datPrimaryRS.Recordset
  .Delete
  .MoveNext
  If .EOF Then .MoveLast
 End With
 Exit Sub
DeleteErr:
 MsgBox Err.Description
End Sub

Private Sub cmdRefresh_Click()
 'This is only needed for multi user apps
 On Error GoTo RefreshErr
 datPrimaryRS.Refresh
 Set grdDataGrid.DataSource =
datPrimaryRS.Recordset("ChildCMD").UnderlyingValue
 Exit Sub
RefreshErr:
 MsgBox Err.Description
End Sub

Private Sub cmdUpdate_Click()
 On Error GoTo UpdateErr

 datPrimaryRS.Recordset.UpdateBatch adAffectAll
 Exit Sub
UpdateErr:
 MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
 Unload Me
End Sub

Private Sub lblLabels_Click(Index As Integer)

End Sub
Private Sub Form_Resize()
 On Error Resume Next
 'This will resize the grid when the form is resized
```

```
 grdDataGrid.Height = Me.ScaleHeight - datPrimaryRS.Height - 30 -
picButtons.Height
End Sub

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

Private Sub datPrimaryRS_Error(ByVal ErrorNumber As Long, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
ByVal HelpContext As Long, fCancelDisplay As Boolean)
 '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:" & Description
End Sub

Private Sub datPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This will display the current record position for this recordset
 datPrimaryRS.Caption = "Record: " &
CStr(datPrimaryRS.Recordset.AbsolutePosition)
End Sub

Private Sub datPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This is where you put validation code
 'This event gets called when the following actions occur
 Dim bCancel As Boolean

 Select Case adReason
 Case adRsnAddNew
 Case adRsnClose
 Case adRsnDelete
 Case adRsnFirstChange
 Case adRsnMove
 Case adRsnRequery
 Case adRsnResynch
 Case adRsnUndoAddNew
 Case adRsnUndoDelete
 Case adRsnUndoUpdate
 Case adRsnUpdate
 End Select
```

```
 If bCancel Then adStatus = adStatusCancel
End Sub

Private Sub cmdAdd_Click()
 On Error GoTo AddErr
 datPrimaryRS.Recordset.MoveLast
 grdDataGrid.SetFocus
 SendKeys "{down}"

 Exit Sub
AddErr:
 MsgBox Err.Description
End Sub

Private Sub cmdDelete_Click()
 On Error GoTo DeleteErr
 With datPrimaryRS.Recordset
  .Delete
  .MoveNext
  If .EOF Then .MoveLast
 End With
 Exit Sub
DeleteErr:
 MsgBox Err.Description
End Sub

Private Sub cmdRefresh_Click()
 'This is only needed for multi user apps
 On Error GoTo RefreshErr
 datPrimaryRS.Refresh
 Exit Sub
RefreshErr:
 MsgBox Err.Description
End Sub

Private Sub cmdUpdate_Click()
 On Error GoTo UpdateErr

 datPrimaryRS.Recordset.UpdateBatch adAffectAll
 Exit Sub
UpdateErr:
 MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
 Unload Me
```

```
End Sub

Private Sub grdDataGrid_Click()

End Sub
Private Const MARGIN_SIZE = 60      ' in Twips
' variables for column dragging
Private m_bDragOK As Boolean
Private m_iDragCol As Integer
Private xdn As Integer, ydn As Integer

Private Sub Form_Load()
   Dim i As Integer
   Dim j As Integer
   Dim m_iMaxCol As Integer

   datPrimaryRS.Visible = False

   With MSHFlexGrid1

      .Redraw = False
      ' place the columns in the right order
      .ColData(0) = 0
      .ColData(1) = 1
      .ColData(2) = 2
      .ColData(3) = 3
      .ColData(4) = 4

      ' loop to re-order the columns
      For i = 0 To .Cols - 1
         m_iMaxCol = i              ' find the highest value starting from this column
         For j = i To .Cols - 1
            If .ColData(j) > .ColData(m_iMaxCol) Then m_iMaxCol = j
         Next j
         .ColPosition(m_iMaxCol) = 0    ' move the column with the max value to the
left
      Next i

      ' set grid's column widths
      .ColWidth(0) = 1905
      .ColWidth(1) = 1695
      .ColWidth(2) = 1755
      .ColWidth(3) = 2250
      .ColWidth(4) = 2130

      ' set grid's column merging and sorting
```

```
    For i = 0 To .Cols - 1
       .MergeCol(i) = True
    Next i

    .Sort = flexSortGenericAscending

    ' set grid's style
    .AllowBigSelection = True
    .FillStyle = flexFillRepeat

    ' make header bold
    .Row = 0
    .Col = 0
    .RowSel = .FixedRows - 1
    .ColSel = .Cols - 1
    .CellFontBold = True

    ' grey every other column
    For i = .FixedCols To .Cols() - 1 Step 2
       .Col = i
       .Row = .FixedRows
       .RowSel = .Rows - 1
       .CellBackColor = &HC0C0C0   ' light grey
    Next i

    .AllowBigSelection = False
    .FillStyle = flexFillSingle
    .Redraw = True

  End With

End Sub

Private Sub MSHFlexGrid1_DragDrop(Source As Control, X As Single, Y As
Single)
'--------------------------------------------------------------------------------
' code in grid's DragDrop, MouseDown, MouseMove, and MouseUp events enables
column dragging
'--------------------------------------------------------------------------------

  If m_iDragCol = -1 Then Exit Sub    ' we weren't dragging
  If MSHFlexGrid1.MouseRow <> 0 Then Exit Sub

  With MSHFlexGrid1
     .Redraw = False
     .ColPosition(m_iDragCol) = .MouseCol
```

```
    .FillStyle = flexFillRepeat
    .Col = 0
    .Row = .FixedRows
    .RowSel = .Rows - 1
    .ColSel = .Cols - 1
    .CellBackColor = &HFFFFFF
    Dim iLoop As Integer
    For iLoop = .FixedCols To .Cols() - 1 Step 2
      .Col = iLoop
      .Row = .FixedRows
      .RowSel = .Rows - 1
      .CellBackColor = &HC0C0C0
    Next iLoop
    .FillStyle = flexFillSingle

    DoSort
    .Redraw = True
  End With

End Sub

Private Sub MSHFlexGrid1_MouseDown(Button As Integer, shift As Integer, X As
Single, Y As Single)
'--------------------------------------------------------------------------------
' code in grid's DragDrop, MouseDown, MouseMove, and MouseUp events enables
column dragging
'--------------------------------------------------------------------------------

  If MSHFlexGrid1.MouseRow <> 0 Then Exit Sub

  xdn = X
  ydn = Y
  m_iDragCol = -1    ' clear drag flag
  m_bDragOK = True

End Sub

Private Sub MSHFlexGrid1_MouseMove(Button As Integer, shift As Integer, X As
Single, Y As Single)
'--------------------------------------------------------------------------------
' code in grid's DragDrop, MouseDown, MouseMove, and MouseUp events enables
column dragging
'--------------------------------------------------------------------------------

  ' test to see if we should start drag
```

```
    If Not m_bDragOK Then Exit Sub
    If Button <> 1 Then Exit Sub                    ' wrong button
    If m_iDragCol <> -1 Then Exit Sub               ' already dragging
    If Abs(xdn - X) + Abs(ydn - Y) < 50 Then Exit Sub   ' didn't move enough yet
    If MSHFlexGrid1.MouseRow <> 0 Then Exit Sub     ' must drag header

    ' if got to here then start the drag
    m_iDragCol = MSHFlexGrid1.MouseCol
    MSHFlexGrid1.Drag vbBeginDrag

End Sub

Private Sub MSHFlexGrid1_MouseUp(Button As Integer, shift As Integer, X As
Single, Y As Single)
'--------------------------------------------------------------------------------
' code in grid's DragDrop, MouseDown, MouseMove, and MouseUp events enables
column dragging
'--------------------------------------------------------------------------------

    m_bDragOK = False

End Sub

Sub DoSort()

    With MSHFlexGrid1
        .Redraw = False
        .Col = 0
        .Row = 1
        .RowSel = .Rows - 1
        .Sort = flexSortGenericAscending
        .Redraw = True
    End With

End Sub

Private Sub Form_Resize()

    Dim sngButtonTop As Single
    Dim sngScaleWidth As Single
    Dim sngScaleHeight As Single

    On Error GoTo Form_Resize_Error
    With Me
        sngScaleWidth = .ScaleWidth
        sngScaleHeight = .ScaleHeight
```

```
    ' move Close button to the lower right corner
    With .cmdClose
        sngButtonTop = sngScaleHeight - (.Height + MARGIN_SIZE)
        .Move sngScaleWidth - (.Width + MARGIN_SIZE), sngButtonTop
    End With

    .MSHFlexGrid1.Move MARGIN_SIZE, _
      MARGIN_SIZE, _
      sngScaleWidth - (2 * MARGIN_SIZE), _
      sngButtonTop - (2 * MARGIN_SIZE)

  End With
  Exit Sub

Form_Resize_Error:
  ' avoid error on negative values
  Resume Next

End Sub
Private Sub cmdClose_Click()

  Unload Me

End Sub

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

Private Sub datPrimaryRS_Error(ByVal ErrorNumber As Long, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
ByVal HelpContext As Long, fCancelDisplay As Boolean)
 '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:" & Description
End Sub

Private Sub datPrimaryRS_MoveComplete(ByVal adReason As
ADODB.EventReasonEnum, ByVal pError As ADODB.Error, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This will display the current record position for this recordset
 datPrimaryRS.Caption = "Record: " &
CStr(datPrimaryRS.Recordset.AbsolutePosition)
End Sub
```

```
Private Sub datPrimaryRS_WillChangeRecord(ByVal adReason As
ADODB.EventReasonEnum, ByVal cRecords As Long, adStatus As
ADODB.EventStatusEnum, ByVal pRecordset As ADODB.Recordset)
 'This is where you put validation code
 'This event gets called when the following actions occur
 Dim bCancel As Boolean

 Select Case adReason
 Case adRsnAddNew
 Case adRsnClose
 Case adRsnDelete
 Case adRsnFirstChange
 Case adRsnMove
 Case adRsnRequery
 Case adRsnResynch
 Case adRsnUndoAddNew
 Case adRsnUndoDelete
 Case adRsnUndoUpdate
 Case adRsnUpdate
 End Select

 If bCancel Then adStatus = adStatusCancel
End Sub

Private Sub cmdAdd_Click()
 On Error GoTo AddErr
 datPrimaryRS.Recordset.MoveFirst
 Exit Sub
AddErr:
 MsgBox Err.Description
End Sub

Private Sub cmdDelete_Click()
 On Error GoTo DeleteErr
 With datPrimaryRS.Recordset
  .Delete
  .MoveNext
  If .EOF Then .MoveLast
 End With
 Exit Sub
DeleteErr:
 MsgBox Err.Description
End Sub

Private Sub cmdRefresh_Click()
```

```
 'This is only needed for multi user apps
 On Error GoTo RefreshErr
 datPrimaryRS.Recordset.MoveNext
 Exit Sub
RefreshErr:
 MsgBox Err.Description
End Sub

Private Sub cmdUpdate_Click()
 On Error GoTo UpdateErr
 datPrimaryRS.Recordset.MoveLast
 Exit Sub
UpdateErr:
 MsgBox Err.Description
End Sub

Private Sub cmdClose_Click()
 Unload Me
End Sub

Option Explicit

' The in-memory database of tips.
Dim Tips As New Collection

' Name of tips file
Const TIP_FILE = "c:\my documents\TIPOFDAY.TXT"

' Index in collection of tip currently being displayed.
Dim CurrentTip As Long


Private Sub DoNextTip()

    ' Select a tip at random.
    CurrentTip = Int((Tips.Count * Rnd) + 1)

    ' Or, you could cycle through the Tips in order

'    CurrentTip = CurrentTip + 1
'    If Tips.Count < CurrentTip Then
'        CurrentTip = 1
'    End If

    ' Show it.
    frmTip.DisplayCurrentTip
```

```
End Sub

Function LoadTips(sFile As String) As Boolean
   Dim NextTip As String   ' Each tip read in from file.
   Dim InFile As Integer   ' Descriptor for file.

   ' Obtain the next free file descriptor.
   InFile = FreeFile

   ' Make sure a file is specified.
   If sFile = "" Then
      LoadTips = False
      Exit Function
   End If

   ' Make sure the file exists before trying to open it.
   If Dir(sFile) = "" Then
      LoadTips = False
      Exit Function
   End If

   ' Read the collection from a text file.
   Open sFile For Input As InFile
   While Not EOF(InFile)
      Line Input #InFile, NextTip
      Tips.Add NextTip
   Wend
   Close InFile

   ' Display a tip at random.
   DoNextTip

   LoadTips = True

End Function

Private Sub chkLoadTipsAtStartup_Click()
   ' save whether or not this form should be displayed at startup
   SaveSetting App.EXEName, "Options", "Show Tips at Startup",
chkLoadTipsAtStartup.Value
End Sub

Private Sub cmdNextTip_Click()
   DoNextTip
End Sub
```

```
Private Sub cmdOK_Click()
   Unload Me
End Sub

Private Sub Form_Load()
   Dim ShowAtStartup As Long

   ' See if we should be shown at startup
   ShowAtStartup = GetSetting(App.EXEName, "Options", "Show Tips at
Startup", 1)
   If ShowAtStartup = 0 Then
      Unload Me
      Exit Sub
   End If

   ' Set the checkbox, this will force the value to be written back out to the registry
   Me.chkLoadTipsAtStartup.Value = vbChecked

   ' Seed Rnd
   Randomize

   ' Read in the tips file and display a tip at random.
   If LoadTips(App.Path & "\" & TIP_FILE) = False Then
      lblTipText.Caption = "That the " & TIP_FILE & " file was not found? " &
vbCrLf & vbCrLf & _
         "Create a text file named " & TIP_FILE & " using NotePad with 1 tip per
line. " & _
         "Then place it in the same directory as the application. "
   End If


End Sub

Public Sub DisplayCurrentTip()
   If Tips.Count > 0 Then
      lblTipText.Caption = Tips.item(CurrentTip)
   End If
End Sub
```

VEHICLE LOAN / RE-IMBURSEMENT INFORMATION SYSTEM

PROJECT WORK DONE AT

HINDUSTAN AERONAUTICS LIMITED,
CORPORATE OFFICE, BANGALORE.

PROJECT REPORT P-770

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

MASTER OF COMPUTER APPLICATIONS

OF BHARATHIAR UNIVERSITY, COIMBATORE

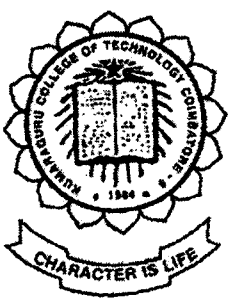
SUBMITTED BY

C.RAVIKUMAR
Reg. No. : 9938M0654

GUIDED BY

EXTERNAL GUIDE
MR.B.KRISHNA KUMAR B.E, M.B.A

INTERNAL GUIDE
MS. P.PARAMESWARI M.C.A



Department of Computer Science & 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 – 641 006

CERTIFICATE

This is to certify that the project work entitled

"VEHICLE LOAN / RE-IMBURSEMENT INFORMATION SYSTEM"

Done By

C. Ravi Kumar
Reg. No: 9938M0654

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

degree of

Master of Computer Applications of Bharathiar University.

S. Thangavel 30/4/02
Professor and Head

Internal Guide

Submitted to University Examination held on 10-05-2002

10/5/02
Internal Examiner

External Examiner

हिन्दुस्तान एरोनाटिक्स लिमिटेड
HINDUSTAN AERONAUTICS LTD.
कारपोरेट आफिस
Corporate Office

4th April 2002

CERTIFICATE

This is to certify that the project entitled **VEHICLE LOAN / REIMBURSEMENT INFORMATION SYSTEM**, being submitted to the Department of Computer Science, Kumaraguru College Of Technology, Coimbatore , by **C.Ravikumar** in partial fulfillment of the requirement for the award of **Master of Computer Applications**, is a bonafide work carried at **Hindustan Aeronautics Limited, Corporate Office Bangalore** from Dec -2001 to Mar- 2002 under our guidance.

Mr. G. Prasanna,
Dy. General Manager(M.S. & Comp.),
Management Services Department,
HAL Corporate Office,
Bangalore – 560 001

B. Krishna Kumar 4/4/02

Mr. B. Krishna Kumar,
Sr. Manager(Computers),
Management Services Department,
HAL Corporate Office,
Bangalore – 560 001

DECLARATION

I hereby declare that the project entitled "***VEHICLE LOAN / RE-IMBURSEMENT INFORMATION 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 **Mr.B.Krishna Kumar B.E, MBA**, Senior Manager (Computers), Hindustan Aeronautics Limited, Corporate Office, Bangalore and **Ms.P.Parameswari MCA**, 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.

C Ravikumar

Place: (C.Ravi Kumar)
Date: Reg. No.: 9938M0654

Countersigned By

(Internal Guide)

ACKNOWLEDGEMENT

First of all, I thank the almighty God and my parents.

I express my profound respect and sincere gratitude to **Dr. K.K. Padmanaban Ph.D,** Principal, Kumaraguru College of Technology, Coimbatore, for his kind co-operation in allowing me tot take up this project work.

I record my sincere thanks to **Dr. S. Thangasamy Ph.D,** Head of the Department, Computer Science and Engineering, Kumaraguru College of Technology, for allowing me to take up the project at **Hindustan Aeronautics Limited, Bangalore**.

I am greatly privileged to express my deep gratitude to my guide **Ms. Parameswari MCA,** Dept of Computer Science and Engineering, Kumaraguru College of Technology, for her valuable advice and encouragement.

I would like to thank **Mr. G. Prasanna, Dy. General Manger (M.S. & Comp.), Hindustan Aeronautics Limited, Bangalore** for giving me an opportunity to undertake this project.

My special and heartful thanks to my beloved guide **Mr. B. Krishna Kumar, Sr. Manager (Computers), Hindustan Aeronautics Limited, Bangalore**. It is he who has shown me the path of creativity and he was more than a guide.

Next, my sincere thanks to all the Staffs of MSD who were kind enough to support me with all necessary support and all other needs.

SYNOPSIS

The Objective of the project entitled Vehicle Loan / Re-imbursement Information System is to develop a software which provides for data entry with all the required checks build in, generation of statistical and analytical reports of the employees working in Hindustan Aeronautics Limited. The aim of this software is to develop an interactive system that can be used by various departments with more efficiency and with more accuracy

The project has three Modules:

I. Interest Subsidy Scheme:

This Scheme is valid to grade officers who are having the grade from 1 to 12.

II. Employee Loan Scheme:

This Scheme is valid to employees who are having the grade from 21 to 33.

III. Re-imbursement Scheme:

This scheme is valid to the officers who are having the vehicles already and not interested in buying new vehicles.

To maintain the data the advanced database Oracle 8i is used and software is being developed using the powerful front - end Visual Basic 6. The Software that is being created is user friendly with all the necessary checks build in.

CONTENTS PAGE NO.

1. INTRODUCTION

1.1 PROJECT OVERVIEW

The project has three Modules:

I. Interest Subsidy Scheme:

This Scheme is valid to grade officers who are having the grade from 1 To 12.They can get Loan from other resources .The company gives subsidy amount according to the officer's grade and type of vehicle.

II. Employee Loan Scheme:

This Scheme is valid to employees who are having the grade from 21 to 33. The company gives advance amount from its own fund according to the employee's type of vehicle.

III. Re-imbursement Scheme:

This scheme is valid to the officers who are having the vehicles already and not interested in buying new vehicles. For these employees, The Company gives allowance according to the grade and type of vehicle.

To maintain the data the advanced database Oracle 8i is used and software is being developed using the powerful front - end Visual Basic 6. The Software that is being created is user friendly with all the necessary checks build in.

1.2 ORGANISATION PROFILE

HISTORY OF H.A.L.GROWTH

Late Shri set up Hindustan Aeronautics Limited. Walchand Hirachand in December 1940, in association with Government of Mysore. The company was registered on 23rd December 1940 as a private company. It was concentrated on the repair and overhaul of military aircraft and engines to support Second World War efforts.

To fulfill the fresh mandate of post independent India, the mission of the company has been redefined as

> "To become a globally competitive aerospace industry, while working as an instrument for achieving self reliance in design, manufacture and maintenance of aerospace defense equipment and diversifying to related areas, managing the business on commercial lines in a climate of growing professional competence."

The Company installed machines and runway and commenced production with the aim of manufacturing

1. HARLOW TRAINER
2. CURTIS HAWK FIGHTER
3. VULTEE BOMBER

In collaboration with INTER _CONTINENTAL AIR CRAFT COMPANY OF U.S.A.

The Government took over the management of H.A.L .in 1945, after Second World War. The activities became

- Reconditioning and Conversion of war supplies.
- Introduction of Dacota for use by civil operations,
- Reconditioning of fighters and bombers for the I.A.F

ORGANISATION OF H.A.L

The H.A.L. is organized into four major complexes. Under each complex several divisions are grouped depending upon its function. They are

BANGALORE COMPLEX

Aircraft,
Engine,
IMGT (Industrial Marine Research & Development Center),
ETBEDC (Engine Test Bed Research & Development Center),
Overhaul,
Barrakpore Division,

Foundry & Forge and

Aerospace Division.

MIG COMPLEX

Nasik and

Koraput Division.

ACCESSORIES COMPLEX

Hyderabad,
Lucknow,
Korwa and
Kanpur Division

DESIGN COMPLEX

Helicopter Design Bureau
Aircrafts Research & Development Center and
Rotary Wing Research & Development Center.

In six decades H.A.L. has spread its wings to cover various activities in the area of design, Development, Manufacture and Maintenance.

The core business of H.A.L. include

1. Design and development of fixed and rotary wing aircraft, Avionics and Accessories.

2. Manufacture, maintenance, repair, and overhaul of :
   -Fighter, Transport and Trainer aircraft
   -Helicopters
   -Aero-Engines
   -Avionics
   -Accessories
   -Ground support equipment

3. Manufacture of structural components for satellites and launch vehicles

4. Software development and Maintenance

5. Design consultancy

FUNCTIONS OF CORPORATE OFFICE

- To develop policy matters for the effective functioning of the company.
- To direct the growth and future development plans of the company.
- To act as a single point contact for the liaisoning with external agencies such a Indian Air Force, Military etc.,

2. SYSTEM STUDY AND ANALYSIS

2.1 EXISTING SYSTEM

The **Vehicle Loan / Re-imbursement Information System** is developed with a view to provide a user friendly atmosphere in which the staff of the P&A department can perform operations on the vehicle database and can provide a statistical and analytical report of employees working in **HAL - Corporate Office.**

The existing system now in **H.A.L Corporate Office** was developed in C, UNIFY (RDBMS) on the Sco - Unix platform (Old –Version).UNIFY (RDBMS) is presently out of the market. So now its maintenance is a major head - ache. The Company has currently implemented a Novell Netware Server and had discontinued with the present Unix server. As a part of the migration plans of the HAL Corporate Office, all the projects currently in the Unix platform are being ported to Novell platform. Moreover, the Unix system presently at work has a character - based Interface and it does not provide a GUI interface to the user. HAL Corporate Office has been planning to build an intra - net and it has a LAN with around 125 PC's connected to it. Hence it was decided to shift to the Novell platform where this networking facility is available with the presentable and user–friendly front – end VB and powerful Back end Oracle. Using this, they can provide information to the concerned persons through the Intra – net.

Procedures Currently In Place

Data Entry:

As a part of the standard MIS, Corporate Office of HAL receives data from the employees with checking some conditions.

Date Storage:

After the data entry, data were stored in the vehicle database.

2.2 PROPOSED SYSTEM

The proposed system is going to be implemented on Novell platform with VB6 as the front end that provides GUI features and MS - ACCESS as the back end data storage. The user can perform any operation, retrieve or provide the required information at his fingertips on the click of a mouse, which was impossible using the existing system.

The Project is implemented in three Modules,

1. Interest Subsidy Scheme
2. Employee Loan Scheme
3. Re – imbursement Scheme

Module I

1. Interest Subsidy Scheme

1.1 Design Issues

The design of this Interest subsidy scheme module includes various options, which are the operations that have to be performed on any database such as adding new records, modifying the existing records, enquiring about records, and deleting unwanted records.

In this Module, two tables are used. One is for storing interest subsidy master data and another is for storing subsidy payment for each employee who is included in interest subsidy master table.

1.2 Algorithms

. Addition

Step 1: Input the Badge Number, Vehicle Type and so on.

Step 2: Process the Input data with some conditions and

This Scheme is valid only once.

So don't allow entering duplicate record.

Step 3: Check all the entries made correctly or not.

Step 4: If it is correct then add to ismaster table.

Modification

Step 1: Input Badge Number of the employee.

Step 2: Traverse through the database

Step 3: Select and display the selected record.

Step 4: Make the required Modification.

Step 5: Save the modified data into the database.

Inquire

Step 1: Input Badge Number of the employee.

Step 2: Traverse through the database.

Step 3: If the key value of Badge Number is present then

Select and display all information about that

particular badge number.

Else

Display the error message.

Deletion

Step 1: Input the Badge Number of the employee.

Step 2: Traverse through the database

Step 3: Select the specified record.

Step 4: Remove it from the database after the confirmation from the user.

Here one more form is used for subsidy payment calculation.

These Subsidy Payments are stored in spmaster table.

Module II

Employee Loan Scheme

In this module, One table is used for storing the application forms of the employees. That is who are applied for the advance.

After received the applications, they are processed with some conditions like, who are applied for the first time, second time and so on. Depending upon the funds, Applications are sanctioned. The sanctioned applications are stored in Vlmaster database and rejected applications are stored in vlreject database.

Then the data from the vlapplication are removed.

The remaining operations are the same as module I. That is Addition, Modification, Deletion and Inquire.

Module III

Re – Imbursement Scheme

In this module, one table is used for storing the option from employee whether he is interested in getting the allowance or reimbursement and another table maintains the data that are applied for the reimbursement option.

Conclusion

These newly designed modules are easy to handle. The user can enter the data with ease as the provision of tab key or enter key has been provided to automatically switch between fields. As all the validations and checks are built-in there is hardly any chance for invalid or missing entries. As it is necessary to keep the information intact, utmost care has been taken in the design of the database

3. PROGRAMMING ENVIRONMENT

3.1 HARDWARE CONFIGURATION

RAM	128 MB
HARD DISC	20 GB
PROCESSOR	Intel Pentium III 833 MHz.
CLOCK SPEED	133 MHz.
TERMINAL	VGA/EGA Monitor
PRINTER	Epson FX 100 (Line Printer)
NETWORKING	Novell Netware 5.0

3.2 DESCRIPTION OF SOFTWARE AND TOOLS USED

Introduction To VB 6.0:

Visual Basic is a High-level language that allows us to create several windows applications easily. It comes in three editions.

- The Learning Edition
- The Professional Edition
- The Enterprise Edition

Depending upon the users application one of the edition is used. Visual Basic provides advanced features such as tools to develop ActiveX and Internet Controls. It also lets the programmers to build distributed application in team environment. VB is a strong front-end tool for Database applications.

Tools And Controls Used In This Project:

Form Designer:

It is the main window in the middle of the screen that lets us to design and edit the application user interface. The Form designer displays two windows for each form. The form itself (the elements of the VUI) and a code window (the code behind the elements of the form).

Multiple Document Interface (MDI):

It is designed to simplify the exchange of information among the documents all under the same roof. The MDI form can't be duplicated but it acts as a container for all other windows & it's called the parent window and the other windows called the child windows. Here in this project one MDI form is used.

ActiveX Data Objects:

ADO provides a minimum number of layers between the front-end application and the data source to provide a high performance interface. It gives a greater control over how an application interacts with the data source and enables us to create more powerful and flexible applications than ADODC. An application trying to access data from a database using ADO involves the creation of three objects- Connection, Command and Record Set.

Label:

This control displays text on the form that the user cannot edit and is mainly used to identify other controls.

Text Box:

It is a mini text editor that a user can edit.

Frame:

This is used to group other controls.

Command Button:

It represents an action that the user carried out when clicks on it.

List Box:

It allows a user to choose an item from a list of items.

Option Button:

Option button provides a set of choices from which a user can select only one button by clicking it at run time.

Menu:

This offers a convenient and consistent way to group commands and an easy way for users to access them.

Common Dialog Control:

This is a custom control that displays the commonly used dialog boxes such as Save As, Color, Font, Print and File open.

Tools Used In This Project:

SQL*Plus:

SQL*Plus is a structured query language supported by Oracle. Through SQL*Plus we can store, retrieve, edit, enter and run SQL commands and PL/SQL blocks. Using SQL*Plus we can perform calculations, list column definitions for any table and can also format query results in the form of a report.

4. SYSTEM DESIGN AND DEVELOPMENT

4.1 INPUT DESIGN

The input design is the important phase in the design of software because the design for handling input data specifies how data are accepted for computer processing. Generally the computer System has intensive interaction with the outside world; mostly this interaction takes place through visual terminals such as monitors. So while developing a software system we have to take care of efficient development of interactive input design. The main objective of our input design is it should be interactive and user friendly.

The quality of input system determines the quality of the system output. Input design features can ensure the reliability of the system and produce results from accurate data.

The Input Design focus on

- Effectiveness - Input forms of screens serve specific purpose
- Accuracy - Design that assures proper completion
- Easy to use - The forms of the screens are straightforward and required no extra time to understand.
- Simplicity - This refers to keeping the forms and screen simple and understandable

Besides knowing what to enter, what should not be entered is equally important. Input design ensures that our software perfectly performs various validations and constant options are filled only with the help of popup selections.

Validation

This is the important factor in input design. The input data is the main source of the system, so proper validation for input data is needed in both field level and form level these are accomplished in our software by including appropriate validation procedures.

Data validation ensures that every value that the user enters into the application is accurate. Visual basic provides two data validation methods:

- Form Level Validation takes place after a user has filled in all fields in the form.
- Field Level Validation takes place as each field on a form is filled in.

When the user types the invalid data into fields on your form, you should provide an audio and/or visual notification of the invalid data entry. With either form level or Field level validation, you should always set focus back to the field that contains invalid data.

4.2 OUTPUT DESIGN

Characteristics of the output system:

- Fit for user needs - The output will produce what the user needs.
- The output should abstract the complexity of the system.
- The output should be accurate and easy to understand.
- Output should be properly formatted.

4.3 DATABASE DESIGN

SCHEME TABLES

Loan Scheme
Vlscheme

Sl_No	Name	Type	Width	Description
1	Vlgrd	Number		Grade / Group
2	Vlyear	Number		Year End March
3	Vlvtype	Number		Vehicle Type
4	Vlloan	Number		Advance Amount
5	Vlnoi	Number		No of Installments

Interest Subsidy Scheme
Isscheme

Sl_No	Name	Type	Width	Description
1	Issgrd	Number		Grade / Group
2	Issyear	Number		Year End March
3	Issvtype	Number		Vehicle Type
4	Isssub	Number		Sub/10000/month
5	Issceil	Number		Ceiling amount

CODE TABLES

Insurance Codes
Inscodes

Sl_No	Name	Type	Width	Description
1	Inscode	Number		Insurance Code
2	Insname	Text	25	Insurance Name

Vehicle Type Codes
Vtpcodes

Sl_No	Name	Type	Width	Description
1	vtpcode	Number		Vehicle Type Code
2	vtpname	Text	20	Vehicle Type Name

Loan Budget
Vlbudget

Sl_No	Name	Type	Width	Description
1	Vlbyear	Number		Year End March
2	Vlbamt	Number		Total Budget Amount

MASTER TABLES

Re-imbursement Master
C_rmst

Sl_No	Name	Type	Width	Description
1	Crpbno	Text	10	Badge Number
2	Crslno	Number		Serial Number
3	Crfrdate	Date/Time		From Date
4	Crtodate	Date/Time		To Date
5	Croption	Text	1	Option (A/R)

Vehicle Loan Master

Vlmaster

Sl_No	Name	Type	Width	Description
1	Vlmpbno	Text	10	Badge Number
2	Vlmslno	Number		Serial Number
3	Vlmyear	Number		Year Ending March
4	Vlmvtype	Number		Vehicle type
5	Vlmadamt	Number		Advance Amount
6	Vlmsurety	Text	10	Badge Number of Surety
7	Vlmnoi	Number		No Of Installments
8	Vlmsdate	Date		Date Of Sanction
9	Vlmddate	Date		Date Of drawl Of Advance
10	Vlmon	Text	1	Old / New Vehicle
11	Vlmpname	Text	25	If Old, Name From Whom Purchased
12	Vlmregno	Text		Registration Number
13	Vlmmodel	Text	25	Vehicle Model
14	Vlmyom	Number		Year Of Manufacturing
15	Vlmvcost	Number		Cost Of Vehicle
16	Vlmmname	Text	25	Manufacturer Name
17	Vlmitype	Number		Insurance Type
18	Vlmicname	Text	50	Insurance Company Name
19	Vlmedins	Date/Time		Expiry Date Of Insurance
20	Vlmcldate	Date/Time		Date of Closure of Loan
21	Vlmnot	Number		No of times Advance taken earlier
22	Vlmflag	Number		Flag of Printing

Vehicle Loan Master

Vlreject

Sl_No	Name	Type	Width	Description
1	Vlrpbno	Text	10	Badge Number
2	Vlrslno	Number		Serial Number
3	Vlryear	Number		Year Ending March
4	Vlrvtype	Number		Vehicle type
5	Vlradamt	Number		Advance Amount
6	Vlrsurety	Text	10	Badge Number of Surety
7	Vlrnoi	Number		No Of Installments
8	Vlrsdate	Date		Date Of Sanction
9	Vlrddate	Date		Date Of drawl Of Advance
10	Vlron	Text	1	Old / New Vehicle
11	Vlrpname	Text	25	If Old, Name From Whom Purchased
12	Vlrregno	Text		Registration Number
13	Vlrmodel	Text	25	Vehicle Model
14	Vlryom	Number		Year Of Manufacturing
15	Vlrvcost	Number		Cost Of Vehicle
16	Vlrmname	Text	25	Manufacturer Name
17	Vlritype	Number		Insurance Type
18	Vlricname	Text	50	Insurance Company Name
19	Vlredins	Date/Time		Expiry Date Of Insurance
20	Vlrcldate	Date/Time		Date of Closure of Loan
21	Vlrflag	Number		Flag of Printing

Interest subsidy master

Ismaster

Sl_No	Name	Type	Width	Description
1	Ispbno	Text	10	Badge Number
2	Isvtype	Number		Vehicle Type
3	Ison	Text	1	New/Old Vehicle
4	Ispname	Text	25	If old, name from whom purchased
5	Ismodel	Text	25	Vehicle Model
6	Isyom	Number		Year of Manufacturing
7	Ismname	Text	25	Manufacturer Name
8	Isvcost	Number		Vehicle Cost
9	Isitype	Number		Insurance Type
10	Isicname	Text	50	Insurance Company Name
11	Isedins	Date/Time		Expiry Date Of Insurance
12	Isbname	Text	25	Source Of Loan(Name of institution.)
13	Islamt	Number		Loan Amount
14	Isldate	Date/Time		Loan Drawn Date
15	Isnoi	Number		No of Installments (months)
16	Isemi	Number		Equated Monthly Installment
17	Isfrdate	Date/Time		Date Of Repayment Of First Emi
18	Iscldate	Date/Time		Date of Closure of Loan

Interest Subsidy Payment Details

Spmaster

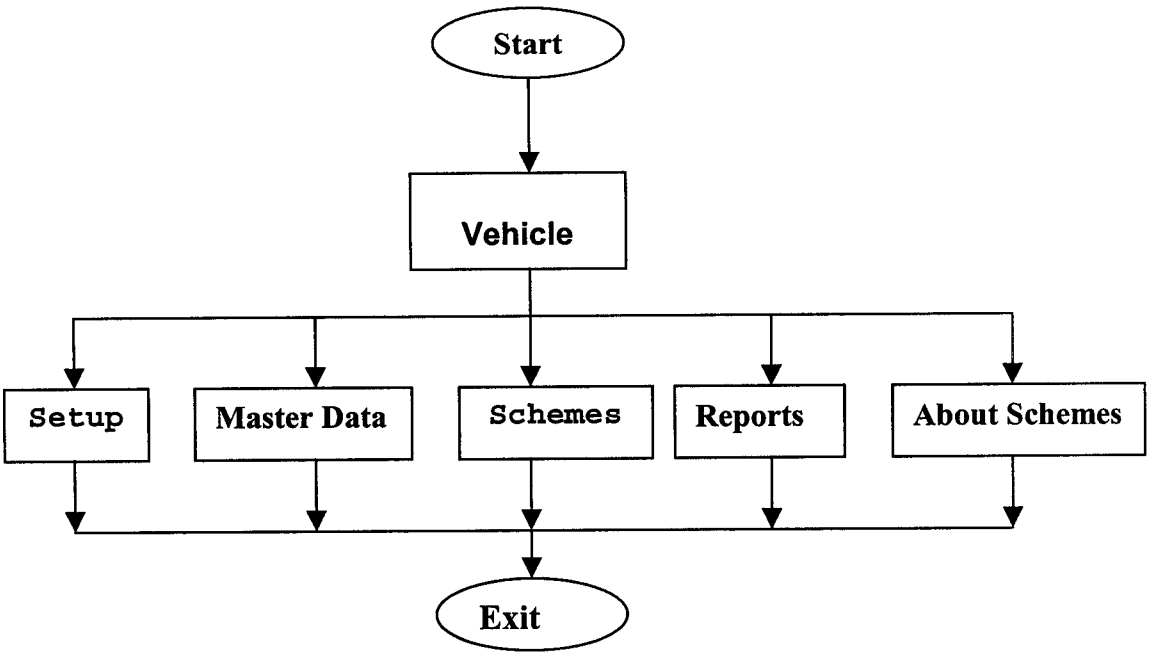
Sl_No	Name	Type	Width	Description
1	Sppbno	Text	10	Badge Number
2	Spyear	Number		Year Ending March
3	Spapr	Number		April
4	Spmay	Number		May
5	Spjun	Number		June
6	Spjul	Number		July
7	Spaug	Number		August
8	Spsep	Number		September
9	Spoct	Number		October
10	Spnov	Number		November
11	Spdec	Number		December
12	Spjan	Number		January
13	Spfeb	Number		February
14	Spmar	Number		March

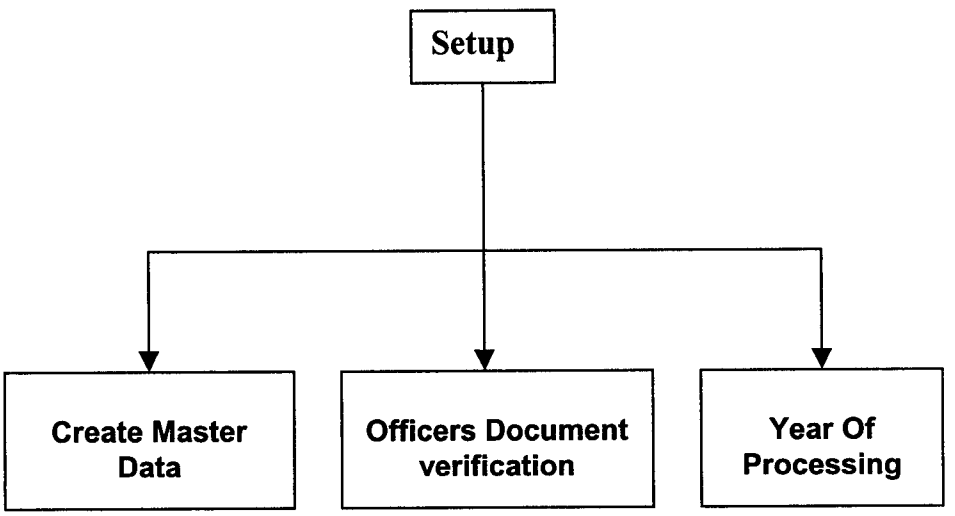
Vehicle Master

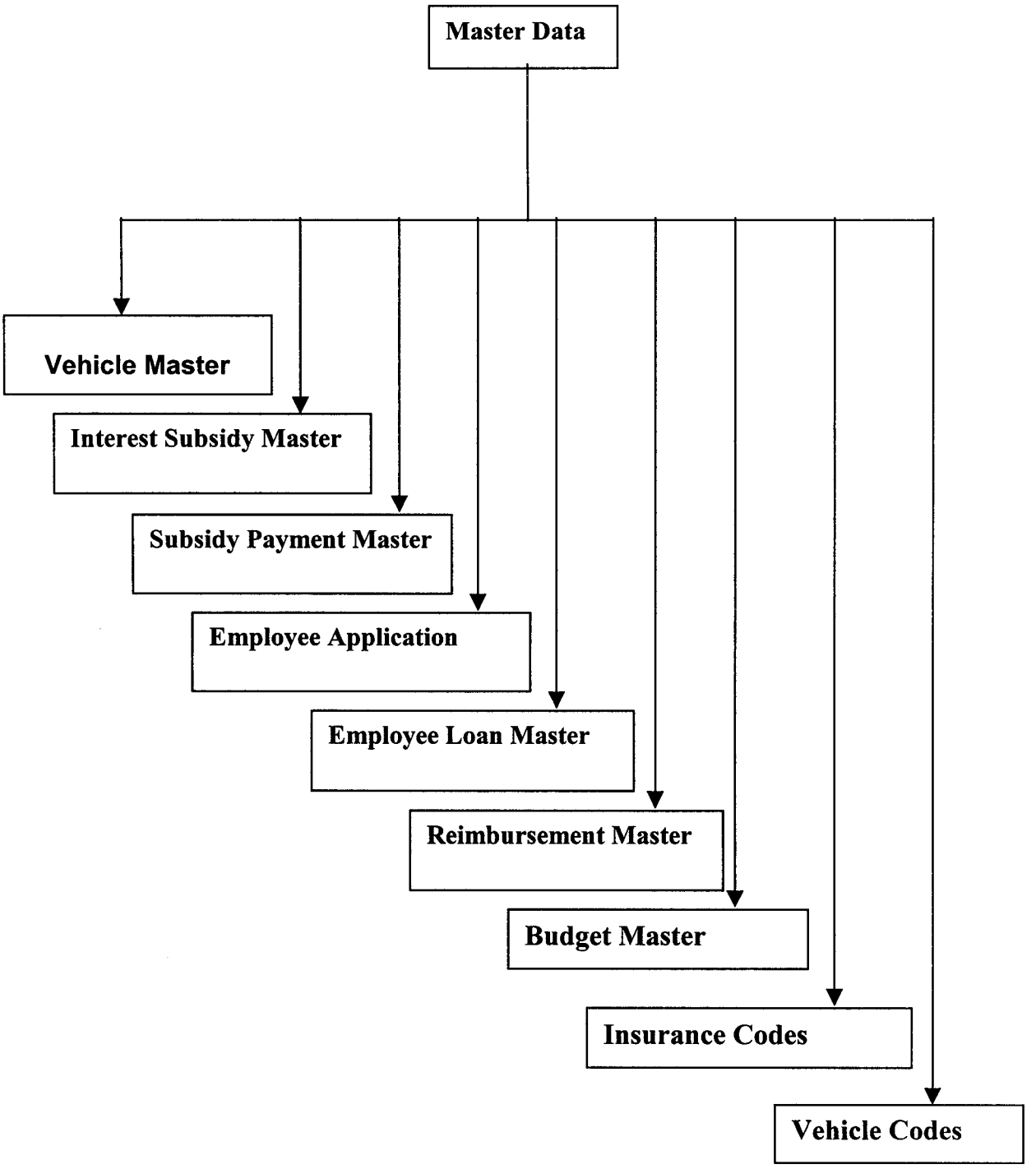
Vehmst

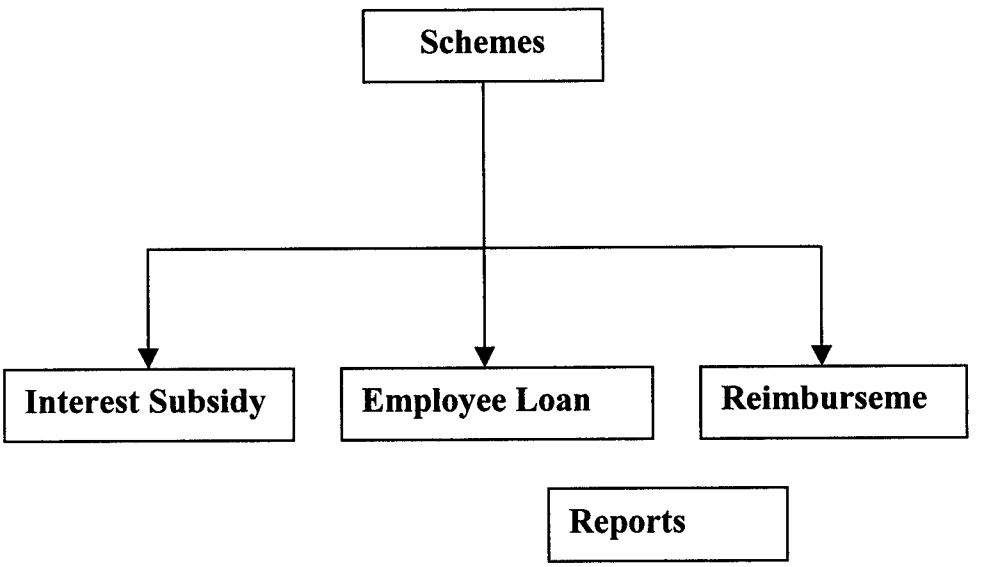
Sl_No	Name	Type	Width	Description
1	Vmpbno	Text	10	Badge Number
2	Vmyear	Number		Year Ending March
3	Vmregno	Text		Registration Number
4	Vmtype	Number		Vehicle type
5	Vmmanf	Text	25	Manufacturer Name
6	Vmmodel	Text	25	Vehicle Model
7	Vmyom	Number		Year Of Manufacturing
8	Vmvcost	Number		Cost Of Vehicle
9	Vmitype	Number		Insurance Type
10	Vmedins	Date/Time		Expiry Date Of Insurance
11	Vmicname	Text	50	Insurance Company Name
12	Vmedtax	Date/Time		Expiry Date Of Tax
13	Vmedreg	Date/Time		Expiry Date Of Registration
14	Vmvdate	Date/Time		Verified Date
15	Vmvby	Text	20	Verified By
16		Number		Flag of Printing

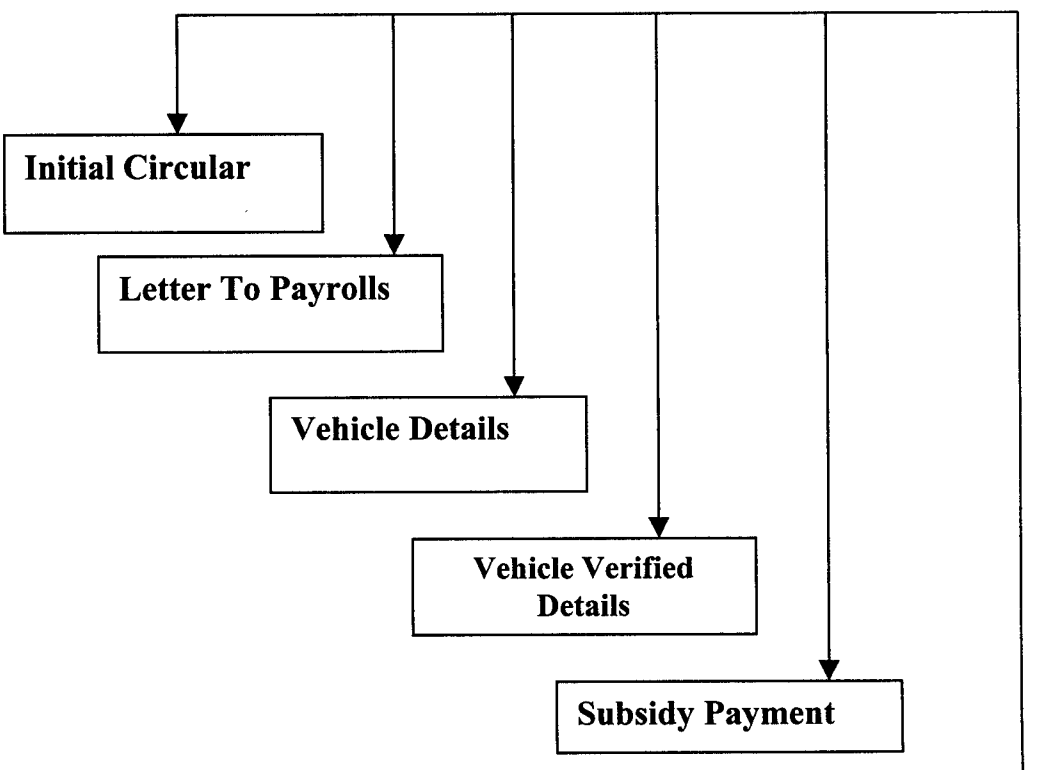
4.4 PROCESS DESIGN






Master Data
Vehicle Master
Interest Subsidy Master
Subsidy Payment Master
Employee Application
Employee Loan Master
Reimbursement Master
Budget Master
Insurance Codes
Vehicle Codes


Schemes
Interest Subsidy
Employee Loan
Reimburseme
Reports


Initial Circular
Letter To Payrolls
Vehicle Details
Vehicle Verified Details
Subsidy Payment

5. SYSTEM IMPLEMENTATION AND TESTING

5.1 SYSTEM IMPLEMENTATION

As a policy every product in the company ready for release undergoes a versioning and release management process. The product is versioned and then implemented in the client location. A complete set of operational documentation, user's manual and guidelines are supplied. Professionals exclusively give user training to a few in the client place from the company.

Implementation Procedures

The project undergoes a versioning and release management before it is delivered to the client. It is a process of identifying and keeping track of different versions and releases of the system. And the released product usually includes Configuration files defining how the release should be configured for particular installations. Data files needed for successful operations. An Installation Program, which is used to help install the system on the target hardware. Electronic and paper documentation describing the system. All these information are made available on a medium, which can be read and understood by the customer for the software.

The following factors are considered before implementation. Checking if all the components which make up the system been included, if the appropriate version of each required component been included, are the data objects included, etc... An installation program is created and the entire kit is delivered to the client.

User Training

The kit delivered consists of a complete guide on the new system developed. A through training on the new system is given to a representative

queries from the audience were answered and hints given on various issues. Special training was given to the admin staff that is to play the role of super user. The configuration details and trouble shooting methodologies were explained and his performance absorbed. The user manual was completely explained and doubts cleared for the same. Installing and uninstalling the package and taking a backup of the data were demonstrated to the super user. Various possible exceptions and the possible causes for it from the user's end were explained. The various user environments and the right of access specified to each user was clearly explained and demonstration given to the team on different user environments. Instructions on successful operation of the system and trouble shooting methodologies were thus discussed.

Operational Documentation

Properly produced and maintained system documentation is a tremendous aid to maintenance engineers. The system documentation includes all of the documents describing the implementation of the system from the requirements specification to the final acceptance test plan.

A complete set of Operational Documentation was prepared for the client, who included the features of the system, the access rights allocated for various users and trouble shooting details. The special features of the system were highlighted. A step-by-step procedure was included in the documentation for data entry, report generation and saving reports in text, html or rich text format. The documentation is prepared keeping in mind users who have little or no knowledge of computers.

5.2 SYSTEM TESTING

Testing Process

Except for small software, systems should not be tested as a single, monolithic unit. Large systems are built out of sub-systems, which are built out of sub-systems, which are composed of procedures and functions. The testing process should therefore proceed in stages where testing is carried our incrementally in conjunction with the system implementation.

There are the five test stages and defects are discovered at any stage, they require program modifications to correct them and this may require other stages in the testing process to be repeated. The process therefore is an iterative one with information being fed back from later stages to earlier parts of the process.

The stages in the testing process are:

- Unit Testing
- Module Testing
- Sub-system Testing
- System Testing
- Acceptance testing

Unit Testing

Individual components are tested to ensure that they operate correctly. Each component is tested independently, without other system components. With respect to this project, the individual functions are treated as component and were tested.

Module Testing

A module is a collection of dependent components such as an object class, an abstract data type or some looser collection of procedures and functions. A module encapsulates related components so it can be tested with

Sub-system Testing

This phase involves testing collection of modules, which have been integrated into sub-systems. Sub-systems may be independently designed and implemented. The most common problems that arise in large software system are sub-system interface mismatches. The sub-system test process should therefore concentrate on the detection of interface errors by rigorously exercising these interfaces. Both the modules are treated as a sub-system and tested in this stage.

System Testing

The sub-systems are integrated to make up the entire system. The testing process is concerned with finding errors, which result from unanticipated interactions between sub-system and system components. It is also concerned with validating that the system meets its functional and non-functional. After integration of the above sub-systems with the whole system, the entire system is tested for errors.

Acceptance Testing

This is the final stage in the testing process before the system is accepted for operational use. The system is tested with data supplied by the system procurer rather than simulated data. Acceptance testing may reveal errors and omissions in the system requirements definition because the real data exercises the system in different ways from testing the data. Acceptance testing may also reveal requirement problems were the system facilities do not really meet the users need or the system performance is unacceptable. Test and reveal data were provided to the system and checked for errors

Defect Testing

Defect testing is intended to exercise a system so that latent defects are exposed before the system is delivered. These contrasts with validation testing which is intended to demonstrate that the system meets its requirement. Validation testing requires the system to perform correctly using given acceptance test cases. A successful defect test is a test, which causes a system to perform incorrectly and hence exposes the defects. It demonstrates the presence, not absence of program faults.

Various values, within the limit and exceeding the limit were provided repeatedly to individual components of data acquisition. These brought out the defects in the system and were corrected.

Two approaches to defect testing are:

Black-box Testing

It relies on the specification of the system or component, which being tested to derive test cases. The system is 'black-box' whose behavior can only be determined by studying its inputs and the related outputs. This is also called as functional testing because mathematical functional can be specified using only inputs and outputs.

Following black-box methods were applied to both the modules to test arrays:

- Usage of only one value of entire array. This proved that the program works for an exceptional array.
- Usage of different arrays of different sizes. This decreased the chances that the program with defect would accidentally produce a correct output because of some characteristic of the

- First, middle and last elements were accessed and any problems due to the boundary effects were delivered.

Structural Testing

This is the complementary approach to black box testing and is sometimes called structural, white-box or glass-box testing. The tester can analyze the code and the use knowledge about the structure of the component to derive test data.

The advantage of structural testing is that an analysis of the code can be used to find how many test cases are needed to guarantee a given level of test courage. A dynamic analyzer can then used to measure the extent of this coverage and help with test case design.

Path Testing

Path testing is a white-box testing strategy whose objective is to exercise every independent execution path through the component. If every independent path is executed then all the statements in the program must have been executed at least one. Furthermore, all conditional statements are tested for both true and false cases. This helped to improve the program efficiency with respect to time complexity and memory usage

6. CONCLUSION

The Vehicle Loan / Reimbursement Information System is developed in such a manner that it is user friendly and menu driven. The GUI screens used for data entry make recording of data much flexible. One can easily operate the system, enter, modify, find, traverse and delete whenever he wants and can generate reports with out much knowledge of computer operations. All operations are simplified in such a way that simply simply pressing key/keys carries them out. All tasks regarding the Vehicle Loan / Reimbursement Information System can be carried out with out much effort. Mostly all problems that are present in the existing UNIX system are eliminated.

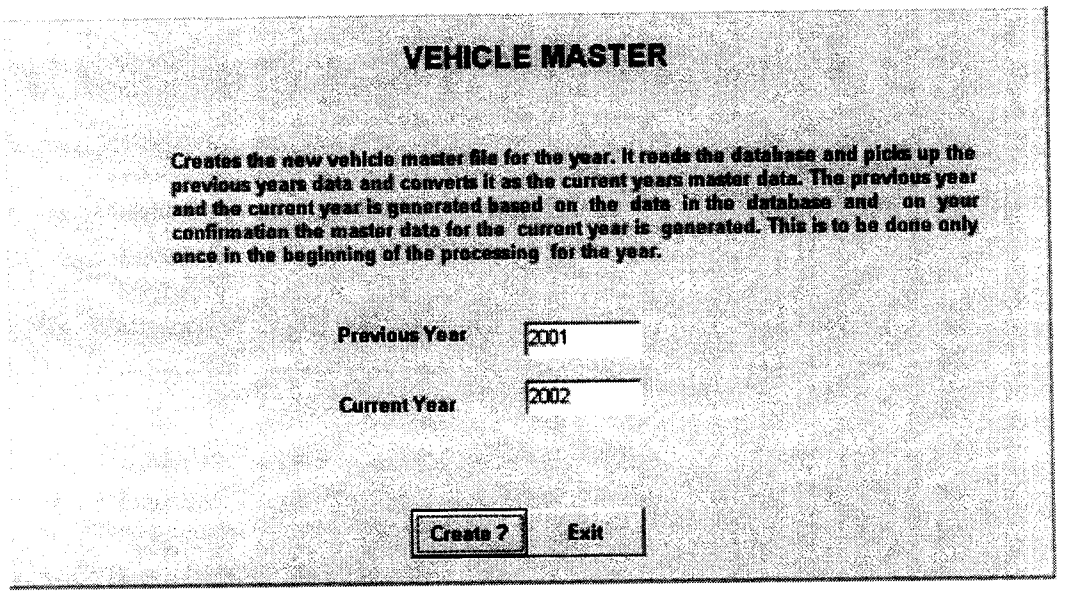
It is possible to enhance the system in future. This system is developed using a modular approach and also the modules are independent of one another. So, in future additional menus and options can be added with no or less modification to the present system.

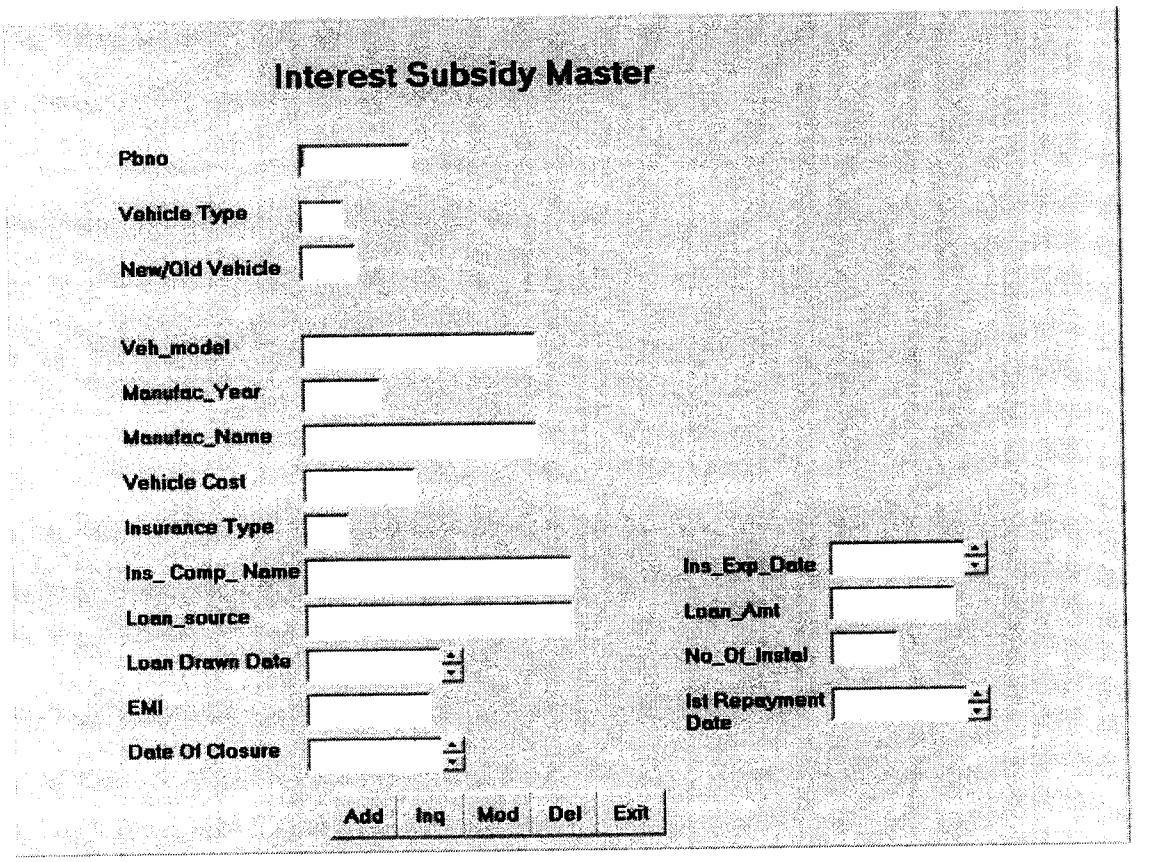
BIBLIOGRAPHY

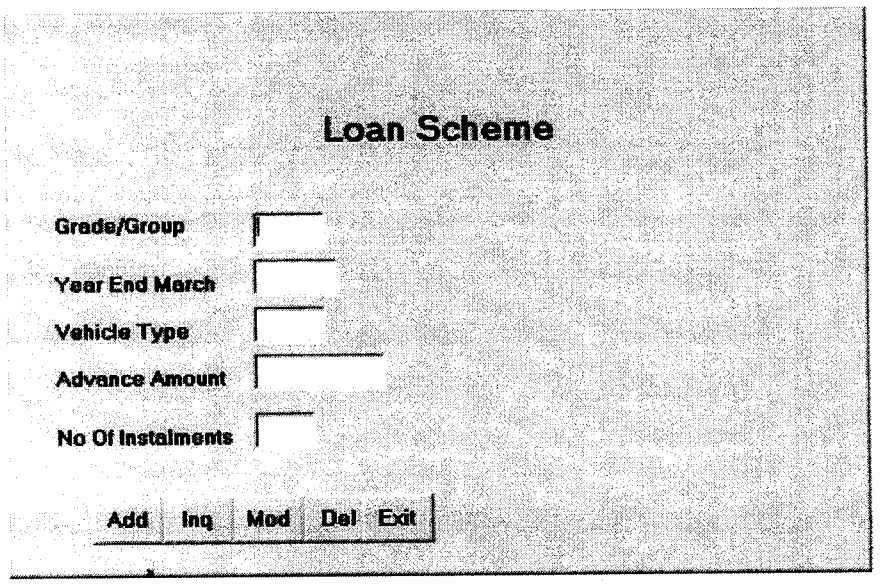
BOOKS	AUTHORS
VISUAL BASIC – 6 In 21 Days	Nathan Gurewich & Ori Gurewich
ORACLE 8i The Complete Reference TMH – Publication	Ivan Bayross
SOFTWARE ENGINEERING CONCEPTS TMH - Publication	Richard Fairly

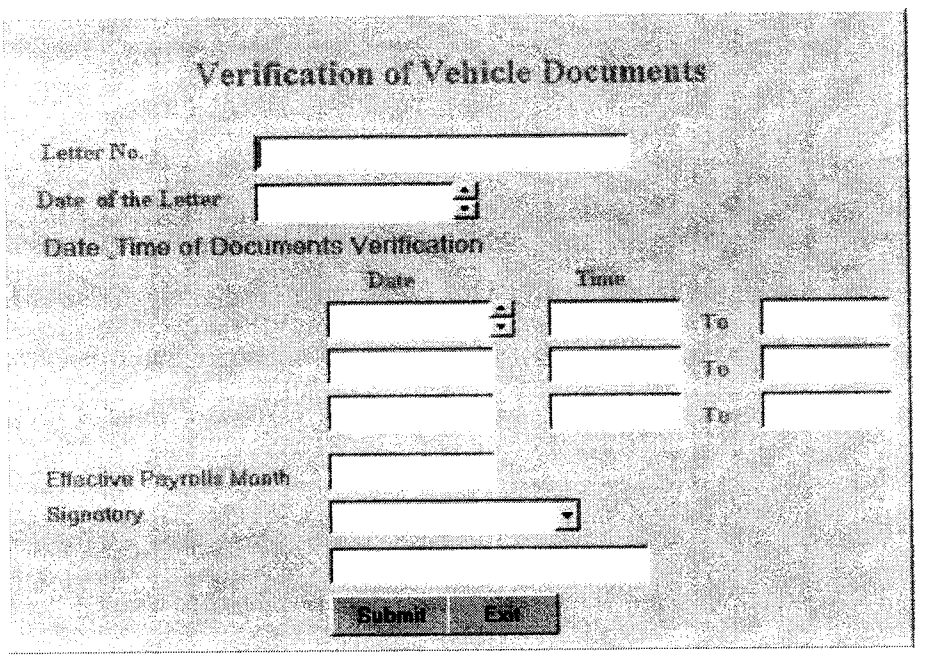
SCREENS

Vehicle Loan / Re-imbursement Information System


VEHICLE MASTER
Creates the new vehicle master file for the year. It reads the database and picks up the previous years data and converts it as the current years master data. The previous year and the current year is generated based on the data in the database and on your confirmation the master data for the current year is generated. This is to be done only once in the beginning of the processing for the year.
Previous Year
2001
Current Year
2002
Create ?
Exit


Interest Subsidy Master
Pbno
Vehicle Type
New/Old Vehicle
Veh_model
Manufac_Year
Manufac_Name
Vehicle Cost
Insurance Type
Ins_ Comp_ Name
Loan_source
Loan Drawn Date
EMI
Date Of Closure
Ins_Exp_Date
Loan_Amt
No_Of_Instal
Ist Repayment Date
Add
Inq
Mod
Del
Exit


Loan Scheme
Grade/Group
Year End March
Vehicle Type
Advance Amount
No Of Instalments
Add
Inq
Mod
Del
Exit


Verification of Vehicle Documents
Letter No.
Date of the Letter
Date ,Time of Documents Verification
Date
Time
To
To
To
Effective Payrolls Month
Signatory
Submit
Exit

Vehicle Verified Documents

Enter the Year

Verified Documents

Print

All Verified Documents

Print

Exit

[redacted]

Class Of Employees Entitled

- *Class Of Employees Entitled*
- *Application For Advance*
- *Amount Of Advance*
- *Execution Of Agreement*
- *Recovery Of Advance*
- *Rate Of Interest*

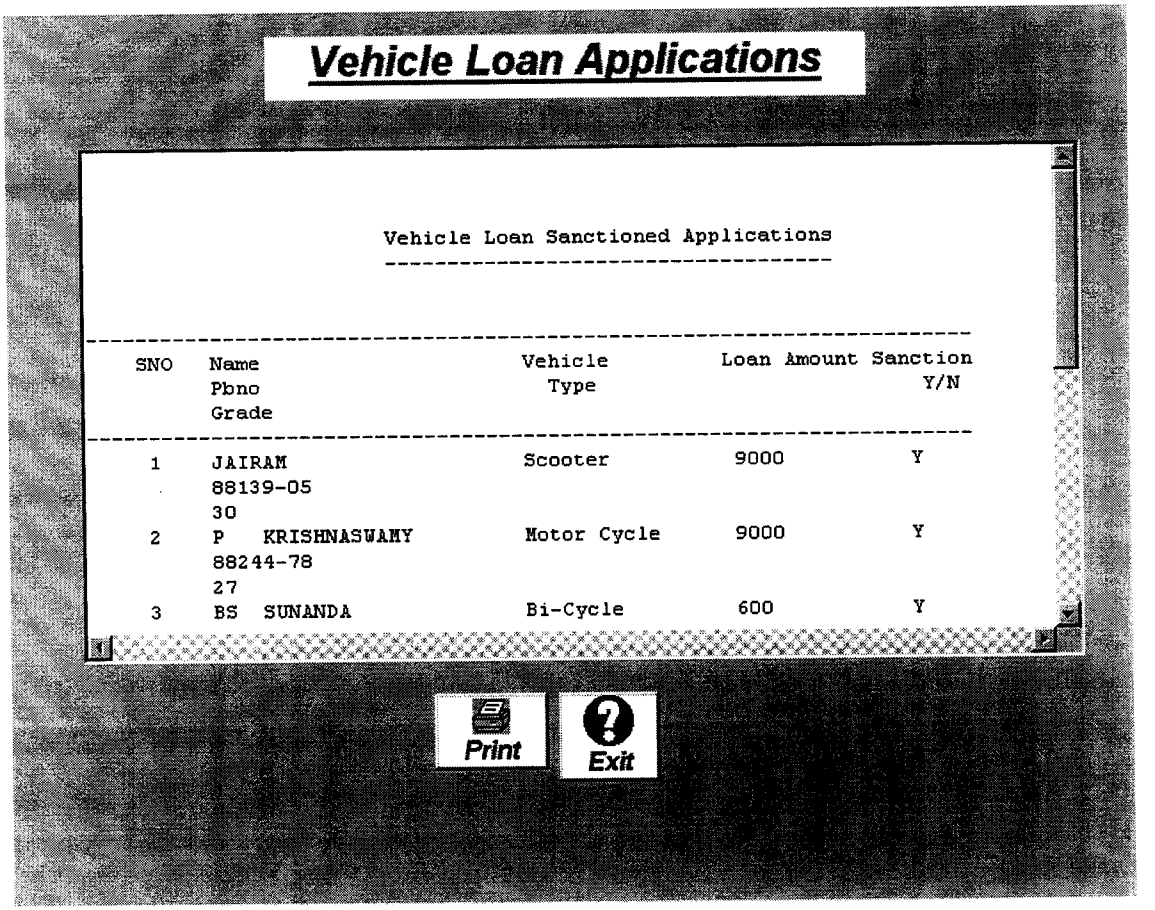
Exit

Class Of Employees Entitled

Permanent employees of the company drawing basic pay of Rs.1377/- and above, in the 1987 scales of pay and [illegible] Scooter will contribute to a more efficient performance of their duties, are eligible for grant of advance for purchase of one of the vehicles mentioned above.

Applications from officers on deputation for purchase of Motor Cycle/ Scooter will be considered in consultation with the parent department of the officer. If that department agrees to ensure recovery of the advance on reversion of the officer, the advance will be granted to him from the Company funds, on the same conditions as would be applicable, if the officer was serving directly under the Government.

If the parent department does not agree to ensure recovery of the advance, the company may consider giving the officer an advance out of the company funds in accordance with the rules applicable to the permanent employees of the company, but recovery of the advance will be so fixed that the entire amount of Advance together with interest there


Vehicle Loan Applications
Vehicle Loan Sanctioned Applications
SNO Name Pbno Grade
Vehicle Type
Loan Amount
Sanction Y/N
1 JAIRAM 88139-05 30 Scooter 9000 Y
2 P KRISHNASWAMY 88244-78 27 Motor Cycle 9000 Y
3 BS SUNANDA Bi-Cycle 600 Y
Print
Exit