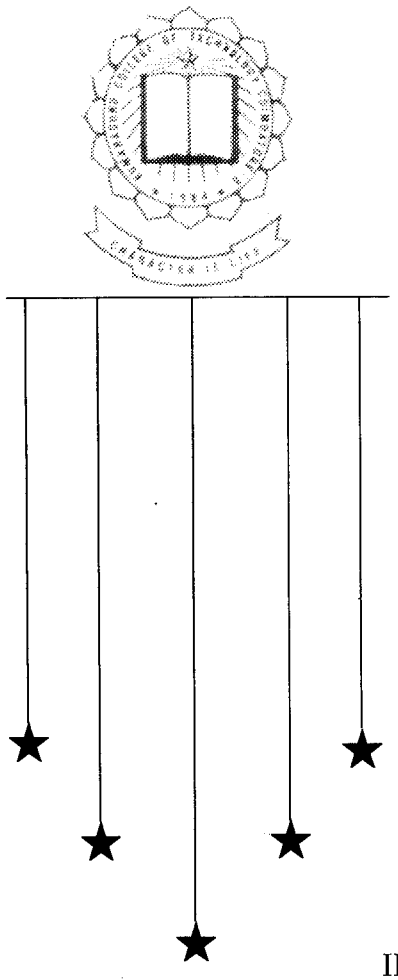
CENVAT PROCEDURE



PROJECT REPORT

P-713

SUBMITTED BY

K. ANGURAJAN (0037Q0026)

GUIDED BY

Mr. S. Mohanavel B.E., M.B.A
Senior Lecturer

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF
MASTER OF SCIENCE IN
APPLIED SCIENCE - COMPUTER TECHNOLOGY
OF THE BHARATHIAR UNIVERSITY, COIMBATORE.

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE – 641006

KUMARAGURU COLLEGE OF TECHNOLOGY

Coimbatore-641006

Department of Computer Science and Engineering

CERTIFICATE

This to certify that the project report entitled

CENVAT PROCEDURE

has been submitted by

Mr./Mrs. K. ANGURAJAN in partial fulfillment of the requirements for the award of degree of Master of Science (Applied Science-Computer Technology) in the Computer Science and Engineering branch of the Bharathiar University, Coimbatore-641046 during the year 2000-2002.

22/4

(Guide)

(Head of the Department)

Certified that the candidate was examined by us in the Project viva-voce examination held on 25-4-2002 and the University Register Number is 0037A0026

(Internal Examiner)

(External Examiner)

DECLARATION

I hereby declare that this project entitled

CENVAT PROCEDURE

submitted in partial fulfillment of the requirement for the award of the degree of M.Sc (Applied Science – Computer Technology) is the report of the original work done by me during the period of study (2001-2002) in

KUMARAGURU COLLEGE OF TECHNOLOGY

COIMBATORE-641006

Under the supervision of

Mr.S.Mohanavel B.E., MBA

Senior Lecturer / CSE Department

Name	Register Number	Signature
K. ANGURAJAN	**0037Q0026**	K. Angu

Place: **Coimbatore**

Date : 22-4-2002

BHARAT HEAVY ELECTRICALS LIMITED

(High Pressure Boiler Plant.)
Tiruchirappalli - 620 014, India.

70-006

Phone No. :

Grams : BHARAT ELEC
Telex : 0455 - 372 & 376
Fax. : 91 -0431 - 553514

Date: 05-04-2002

PROJECT COMPLETION CERTIFICATE

This is to certify that Mr. K. ANGURAJAN Register number 0037Q0026, Final Year M.Sc (Applied Science-Computer Technology) of Kumaraguru College of Technology, Coimbatore has completed the project work on "CENVAT PROCEDURE" from January 01-01-2002 to March 31-03-2002 as partial fulfillment of his Master of Science (Applied Science-Computer Technology) degree.

The project code is confidential and sole property of Bharat Heavy Electricals Limited and the same cannot be reproduced in any form (software copy / hardware copy).

During the course of the project his conduct was good.

19/4/02

Bharat Heavy Electricals Limited

M. SHANTHI,
ACCOUNTS OFFICER (EXCISE)
BHEL., TRICHY-620 014

ACKNOWLEDGEMENT

I express my deep sense of gratitude to our principal Dr.K.K.Padmanabhan B.Sc (Engg.), M.Tech., Ph.D., for having provided necessary facilities for the successful completion of my project.

I also extend my sincere thanks to Dr.S.Thangasamy Ph.D., Professor and Head of Computer Science and Engineering, for the help rendered by him to complete my project successfully.

I give my immense pleasure to express heartfelt thanks to my guide Mr.S.Mohanavel, B.E., MBA, Senior Lecturer, Department of Computer Science and Engineering, for his encouragement and valuable suggestions to make this project a successful one.

I express my sincere thanks to Mrs. Shanthi, M.Com., ICWA, Accounts Officer, Bharath Heavy Electrical Limited, Trichy for permitting me to do the project work and providing all the facilities to make the project a success.

I am extremely thankful to my Class Advisor Mr. R. Dinesh, M.S. Senior Lecturer and the Project co-ordinator Mr. K.Sivan Arul Selvan, M.Sc., M.C.A., Lecturer for their kind support, encouragement and valuable suggestions for completing the project.

Last but not least I extend my heartfelt thanks to all the faculty, friends and well wishers who helped me in completing this project work.

CENVAT PROCEDURE

The project titled "CENVAT PROCEDURE" is developed for Bharath Heavy Electricals Limited (BHEL) Trichy. It is an excise duty relief scheme where the manufacturer of the product can avail credit on the duty paid and can set-off against the duty payable on the final product.

Central Excise section of BHEL, collects excise duty from suppliers, consolidates all the documents obtained from stores, and makes an entry in MODVAT (Modified Value Added Tax) receivable account. Quality Control department approves the materials sent by suppliers after inspection and raises Stores Receipt Voucher for payment by bank. Bank makes a debit entry in MODVAT receivable account. The problem is that the credit and debit entries made in MODVAT receivable Account does not match.

The project aims at matching the two entries based on fields common in both entries. The matched records are stored in separate tables. The entries are text files, which are received from stores and bank using File Transfer program. Chat program is developed to achieve communication amongst the departments. The application has facilities to view, delete and query the tables. Reports are provided to take prints and export as other files.

The project is developed using Visual Basic as Front-end tool and Ms-Access as Back-end tool.

CONTENTS

1.1 About the Project

The project includes three departments of BHEL stores, bank and central excise department. Materials supplied from vendors reaches stores and the entries made are sent as text files to central excise section. Invoices are sent to central excise section accounts officer who collects excise duty for the goods supplied from vendors. Simultaneously vendor sends the invoice to banking for payment.

Excise duty is the main source of income to central government. Each supplier must pay excise duty for their goods. The purpose of the project is to find out whether the Central Excise Section has collected the Excise duty for the materials supplied by the vendors and payment is made to the supplied goods. This is done by matching the bank and stores records.

Only 90% of the records will get matched during reconciliation process and the remaining records have to be handled manually because the matching fields purchase order number and invoice number don't have entries or the excise duty value would be incorrect. Few reasons for incorrect excise duty value is

- Materials would have been damaged during transportation.
- Supplied goods would be less than the total goods ordered.
- Weather conditions such as evaporation of petrol.
- unreachable goods due to accidents and other natural calamites.

Being a large scale industry it is impossible to make the entire process streamlined.

The main features of this project comprises the following

- User friendly
- Security is provided for both the application and database.
- Communication amongst the departments is achieved by chat application.
- File Transfer program is used to obtain text files from the stores and bank.
- Text files are appended to appropriate tables.
- Application provides retrieving of selected fields through queries, also to view and delete tables.
- Matched records are displayed in Data Grid and stored in separate tables.
- Reports are provided for tables with print and export facilities.

For all the above purpose the users are provided with a good user interface by which user can move through it.

The proposed system should be developed in such a way as to solve the above problems faced by the present normal system by using latest technologies. For this, the ideal environment is the windows environment; as it's the most popular multitasking system available today. The tool used to develop the system is Visual Basic for its flexibility and versatility.

The proposed system uses Microsoft Access that is a database system with powerful querying functions and easy accessing. The system is very user friendly and driven by menus. Enough securities are provided to the databases and the new system. A good user interface is provided to the user. File transfer between the clients and Chatting service facilitates the user to communicate between them.

1.2 Bharath Heavy Electricals Limited

BHEL is the largest engineering and manufacturing enterprise in India in the energy related/infrastructure sector today. BHEL was established more than 40 years ago when its first plant was set up in Bhopal ushering in the indigenous Heavy Electrical Equipment industry in India, a dream which has been earning profits continuously since 1971-72 and achieved a sales turnover of Rs.6347 crores with a pre-tax profit of Rs.294 crores in 2000-2001.

Industries

BHEL is a major contributor of equipment and systems to industries: cement, sugar, fertilizer, refineries, Petrochemicals, paper, oil and gas, metallurgical and other process industries. The range of systems & equipment supplied includes : captive power plants, co-generation plants, DG power plants, industrial steam turbines , industrial boiler and auxiliaries, waste heat recovery boilers, gas turbines, heat exchanges and pressure vessels, centrifugal compressors, electrical machines, pumps, valves, seamless steel tubes, electrostatic precipitators, fabric filters, reactors, fluidized bed combustion, chemical recovery boilers and process controls.

Transportation

BHEL is involved in the development design, engineering, marketing, production, installation, and maintenance and after-sales service of rolling stock and traction propulsion systems. In the area of rolling stock, BHEL manufactures electric locomotives up to 5000 HP, diesel-electric locomotives from 350 HP to 3100 HP, both for mainline and shunting duty applications. BHEL is also producing rolling stock for special applications vis., overhead equipment cars, Special well wagons, Rail-cum-road vehicle etc.

International operations

BHEL has, over the years, established its references in around 60 countries of the world, ranging from the United States in the west to New Zealand in the Far East. These references encompass almost the entire product range of BHEL, covering turnkey power projects of thermal, hydro and gas-based types, substation projects, rehabilitation projects, besides a wide variety of products, like transformers, insulators, switchgears, heat exchangers, castings and forgings, valves, well-head equipment, centrifugal compressors, photo-voltaic equipment etc. Malaysia and execution of four prestigious power projects in Oman. Some of the other major successes achieved by the company have been in Australia, Saudi Arabia, Libya, Greece, Cyprus, Malta, Egypt, Bangladesh, Azerbaijan, Sri Lanka, Iraq etc.

Human Resource Development Institute

The most prized asset of BHEL is its employees. The Human Resource Development Institute and other HRD centers of the company help in not only keeping their skills updated and finely honed but also in adding new skills, whenever required. Continuous training and retraining, a positive work culture and Participate style of management, have engendered development of a committed and motivated workforce, leading to enhanced productivity and higher levels of quality.

Health, Safety and Environment Management

BHEL, as an integral part of business performance and in its endeavor of becoming a world class organization and sharing the growing global concern on issues related to Environment, Occupational Health and Safety, is committed to protect Environment in and around its own establishment, and to providing safe and healthy working environment to all its employees.

2.1 Problem Definition

Excise duty is the main source of income to the central government. The percentage that is to be collected as excise duty is levied on the basis of annual income obtained by the supplier. It is 9% if the annual income is less than 3 crores, and 16% if it is 3 crores and above. The excise duty is collected on the total price of the goods provided by the suppliers.

All departments requirements are sent to the purchase department who then assigns it to selected vendors. Materials reach stores and the stores send a document to Central Excise section. Central Excise consolidates all the entries and takes credit and enters in MODVAT (Modified Value added Tax) receivable account. Quality department approves the materials after inspection and raises Stores Receipt Voucher for payment by bank. Bank makes a debit entry in to MODVAT receivable account.

The problem is that the credit and debit entries do not match correctly. The reason is

- If a supplier sends goods for Rs.100, then 16% of Rs.100 i.e. 16 Rs is collected as excise duty. When the goods are found to be worth for Rs.75 due to damage or any reasons by quality department then it's a loss to supplier.

- If a supplier sends goods for Rs.100, before check by the quality department supplier collects money for the supplied goods from the bank. Excise Department would have not collected Duty for the supplied goods. Then it is a loss to the excise department.

2.2 Existing system

Requirement analysis is used to analyze the knowledge about the existing system. Reconciliation is the process of matching the stores and bank records to determine whether the Central Excise department has collected the Excise duty and the supplier is paid for the goods received. Stores and Bank sent the text files to server and Central Excise department transmits to its system to perform reconciliation. There is six different set of matching performed and for each matching a separate program is written by the Accounts Officer.

2.2.1 Limitations

- No security for the text files since any client can access the server.
- Communication between the departments is less due to limited phone facilities and large distances.
- Reconciliation programs are not efficient and include a lot of work.
- Database can be easily accessed and modified which leads to lot of confusion.
- Only the person who developed it can use the application.
- Wastage of Time.

2.3 User Characteristics

The client or user using the system should have the following knowledge.

- The user should atleast have a knowledge about Visual FoxPro.
- The user should know to retrieve files from server and convert it into tables.
- The user should know to generate reports for prints.

2.4 Proposed system

The proposed system is a menu driven application which overcomes all the above stated limitations. It makes use of four modules with additional File transfer and Chat facilities.

The first module is **Login menu** with connect and exit menu items. The connect form gets login name, password and if it's valid a database login form appears. It gets path name and password of the database and checks for validity. If valid the next menus are enabled else prompts the user with the same form. The user can exist from the application at any time by selecting exit menu item.

The second module is Table menu with four menu items.

- Append
  - Stores
  - Append Bank
    - Bank
    - months
- View
- Delete
- Query

In **Append form** the text files of bank and stores are converted in to bank and stores tables. Months is used to insert the month on which reconciliation is performed in to bank table.

In **View form** all the tables can be viewed.

In **Delete form** all the records of tables can be deleted after getting confirmation for user.

In **Query form** user can execute his own queries.

The third module is reconciliation menu with six matching

- Purchase Order Number, Invoice Number and Basic Excise Duty
- Purchase Order Number, Invoice Number and Basic Excise Duty +/ - Rs.100
- Purchase Order Number and Basic Excise Duty
- Purchase Order Number and Invoice Number
- Invoice Number and Basic Excise Duty
- Unmatched

The fourth module is Reports menu to take prints for all the table records and export them as different files.

File Transfer Module

It is a client to client file transfer application without the use of server. One client must be in a listening mode waiting for the connection and on receiving the connection request form another client both the clients can transfer only text files between them. The connection information is displayed in both clients text box and knows which one gets disconnected.

Chat Module

In chat application all the clients must connect to the server in order to chat with other clients. Server maintains a client list in which a client gets added or removed automatically based upon connect or close operations from the client. Client can transmit message to a particular client or to all connected to the server. Each client can obtain the currently available clients list from the server by a button click. Messages are transmitted between the clients only through the server.

2.5 Feasibility study

The project is to be considered feasible only if the proposed system is useful. Thus the purpose of the feasibility is "to gather, analyze and document the data needed to make an informal intelligence decision regarding a system's practicability".

Technical Feasibility

Using the Visual Basic along with windows as the operating system and the back-end as MS-Access the project is developed. The memory capacity and speed of the existing hardware is quite sufficient for the execution of the system. Technical enhancement may be needed in this system in future, and it will not pose any barriers to estimated budget. Thus a thorough study reveals that this project is technically feasible.

Economic Feasibility

The cost of the system is evaluated here. There is no extra cost needed for implementing the system, because this organization already has a LAN facility, network and windows environment. Since it is very easy to use, no training is needed. So training cost can be avoided. This system is flexible so that further enhancement is possible according to the future needs of the users.

Operational Feasibility

This system provides all the facilities and can be used with the windows environment. The operation of the system is not complicated for the users.

3. Environment Description

3.1 Visual Basic

Visual Basic is an ideal programming language for developing sophisticated professional applications for Microsoft Windows. It makes use of Graphical User Interface for creating robust and powerful applications. Features such as easier comprehension, user-friendliness, faster application development and many other aspects such as introduction to ActiveX technology and Internet features make Visual Basic an interesting tool to work with.

One of the most significant changes in visual Basic is the Integrated Development Environment (IDE). IDE is called integrated because we can access virtually all of the development tools that we need from one screen called an interface. The IDE is also commonly referred to as the design environment, or the program. The Visual Basic IDE is made up of a number of components

In previous versions of Visual Basic, the IDE was designed as a Single Document Interface (SDI). In a Single Document Interface, each window is a free-floating window that is contained within a main window and can move anywhere on the screen as long as Visual Basic is the current application. But, in Visual Basic 6.0, the IDE is in a Multiple Document Interface (MDI) format. In this format, the windows associated with the project will stay within a single container known as the parent. Code and form-based windows will stay within the main container form.

Winsock Control

A WinSock control allows you to connect to a remote machine and exchange data using either the User Datagram Protocol (UDP) or the Transmission Control Protocol (TCP). Like the Timer control, the WinSock control doesn't have a visible interface at run time.

Selecting a Protocol

When using the WinSock control, the first consideration is whether to use the TCP or the UDP protocol. The TCP protocol control is a connection-based protocol, the user must establish a connection before proceeding. The UDP protocol is a connectionless protocol, message is sent from one computer to another, but there is no explicit connection between the two.

TCP Connection Basics

When creating an application that uses the TCP protocol, you must first decide if your application will be a server or a client. Creating a server means that your application will "listen," on a designated port. When the client makes a connection request, the server can then accept the request, completes the connection, and the client and server can freely communicate with each other.

UDP Basics

Creating a UDP application is even simpler than creating a TCP application because the UDP protocol doesn't require an explicit connection.

About the Bind Method

The Bind method is invoked when creating a UDP application. The Bind method "reserves" a local port for use by the control. This may come in useful if you wish to prevent another application from using that port.

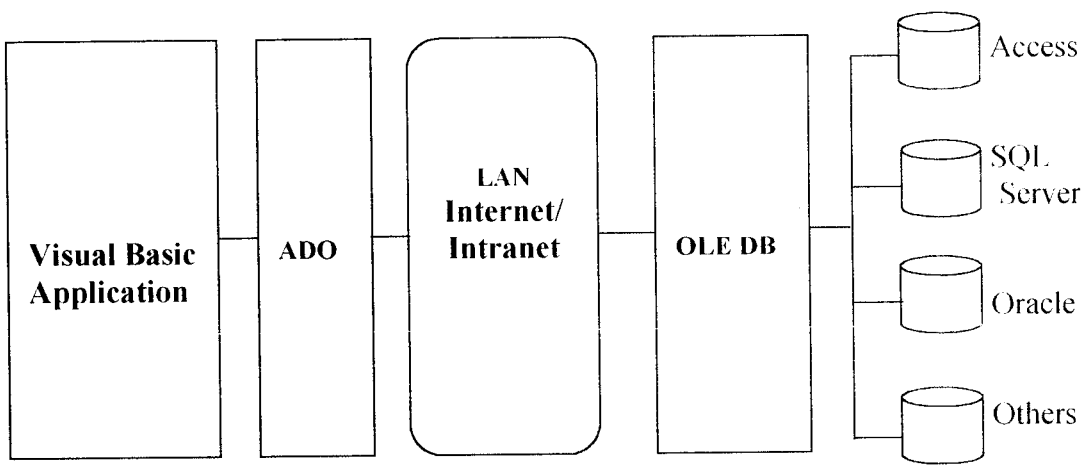
P-713

Overview of ADO

The goal of Universal Data Access is to let us build enterprise-wide applications that can access a variety of relational and non-relational data across any type of network.

ADO Object Model

ADO uses a new database connection framework called OLE DB, which allows faster, more flexible access to multiple data providers, and ADO wraps it all into one easy-to-use interface. This means that we can write database applications that can easily scale from single-user databases such as Access to complex client server systems such as Oracle, SQL server, or almost any other database that is either an OLE DB provider, or ODBC provider.



Connection

The ADO Connection object represents an open connection to a data source. The Connection object specifies all the necessary parameters, the server and the database names, the user's name and password, and the timeout—before opening the data source.

Command

The ADO Command object defines a command or a query that can execute on a data source.

Record Set

The Record set object contains all the data fetched from a database or to send to it. A Recordset can include several rows and columns of data.

Field

The Field object exists as part of recordset and contains information about a single column of data within a recordset.

Parameter

A Parameter object represents a parameter in the parameterized command or stored procedure on which a Command object is based.

Property

The Connection, Recordset, Command, and Field objects expose Properties collections, which contain all the dynamic properties that the ADO provider has added to the built-in properties that are referenced using the standard dot syntax.

Data Environment and Data Report

Data environment designer is a powerful, sophisticated interface, which provides an interactive, design time environment for creating programmed runtime access to data.

Data View Window

After the Data Environment Designer is created, objects can be dragged from data view window to the Data Environment Designer.

Data Report

Data report utility introduced for the first time in 0Visual Basic 6.0 is used to design reports. In addition to creating printable reports, the report can be exported to HTML or text files.

3.2 Ms-Access

Ms Access does things differently from database programs like dbase for dos. Access stores records in organized lists called tables. One or more tables in access take the whole database.

A table is just a collection of records with a same structure. All the records in the tables contain the same type of information. Access allows to setup tables and links them to other tables.

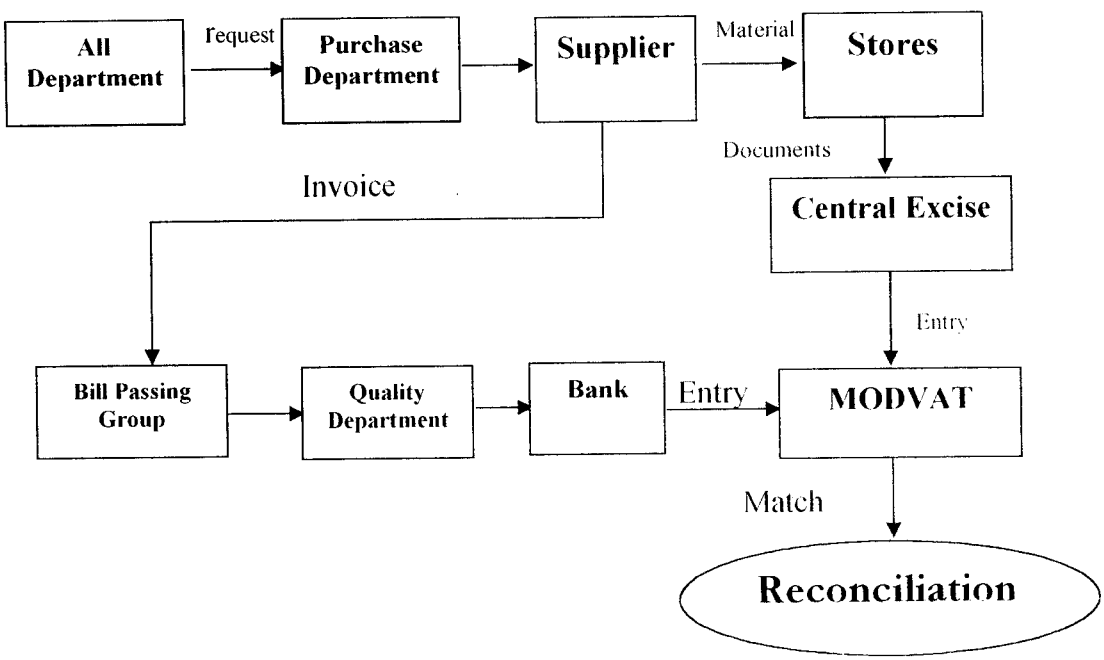
Ms Access is a relational database. This means that a data in a table is linked through one or more fields present in other tables. In this feature of linked tables that separates database programs like Ms Access and other types of data base. For this purpose of database connection only three properties are used along with the data control objects. They are connected by the database home and Record source. Ms Access is the database system used in this project. Everything in ms access fields, records and tables are called objects. In Ms-Access tables contain records and fields .The tables are then stored in a single database file. All the objects are kept together and seem a lot better.

SQL

Structure query language is a set of commands that lets YOU access a relational database. SQL has a simple command structure for data definitions, data access and data Manipulation. SQL is set oriented. Sql is non procedural. When we use SQL we specify what has to be done rather than how to do it? To access data we need to name a table and the columns, we don't have to describe on access method.

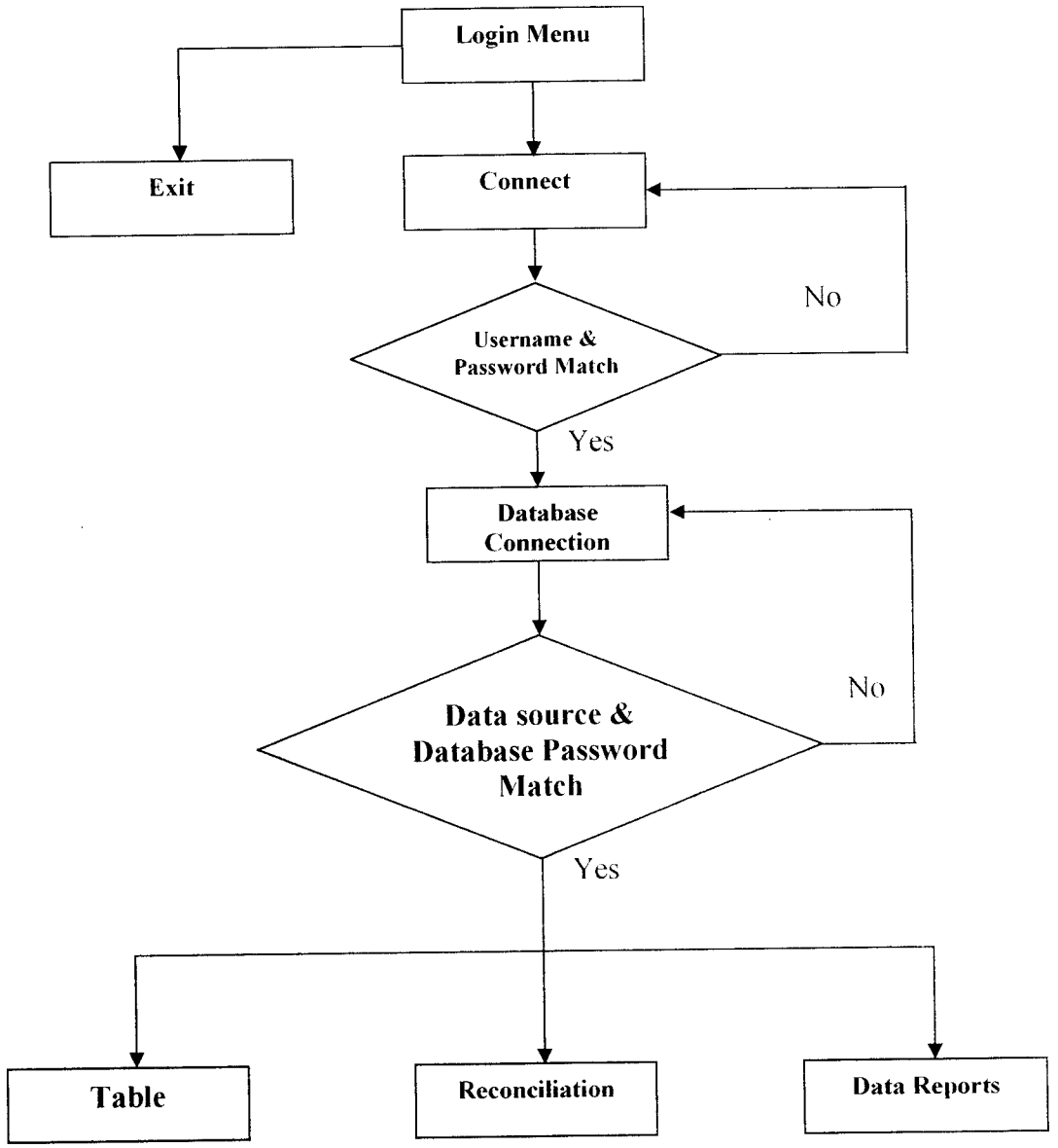
4. System Design

Data Flow Diagram



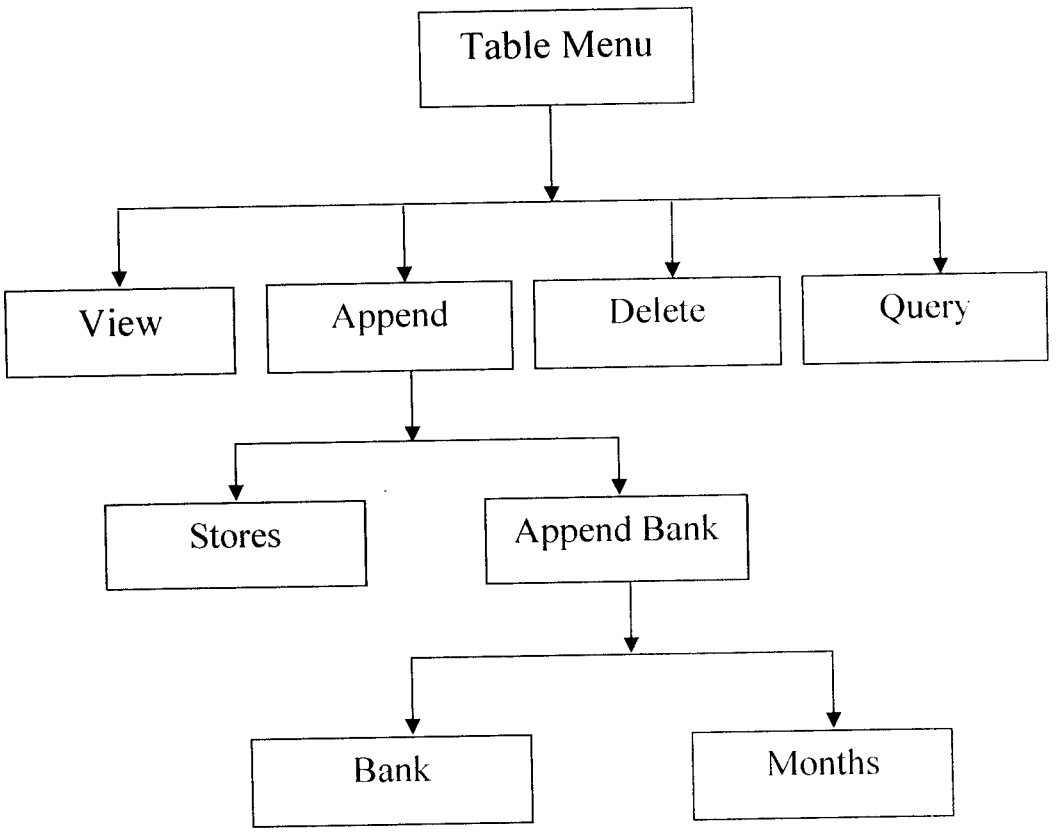
All department requirements are send to purchase department and they select a supplier through tenders. Materials reaches stores and the documents are transferred to central excise section where they make an entry into MODVAT (Modified Value Added Tax) account. Simultaneously the supplier sends invoice slip to bill passing group. Quality department inspects the materials and allows bank for payment for the supplied goods. Bank makes an entry into MODVAT (Modified Value Added Tax). Both the entries are checked for reconciliation.

Login Form



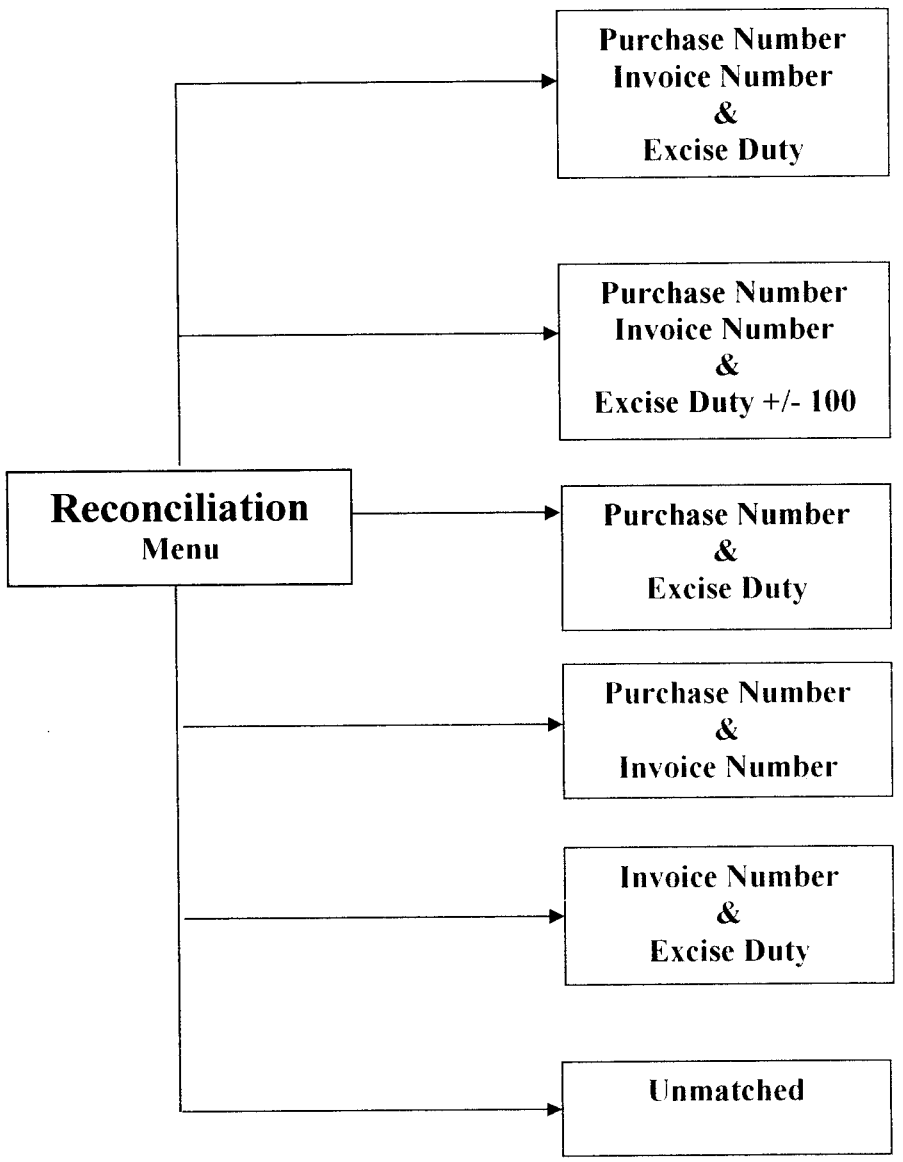
In login menu, connect menu item is chosen. It prompts for username and password. If it matches, then database connection form is displayed. It prompts for path and password of database. If it matches, then table, reconciliation and data reports menu will be enabled else prompts database form.

Table Form



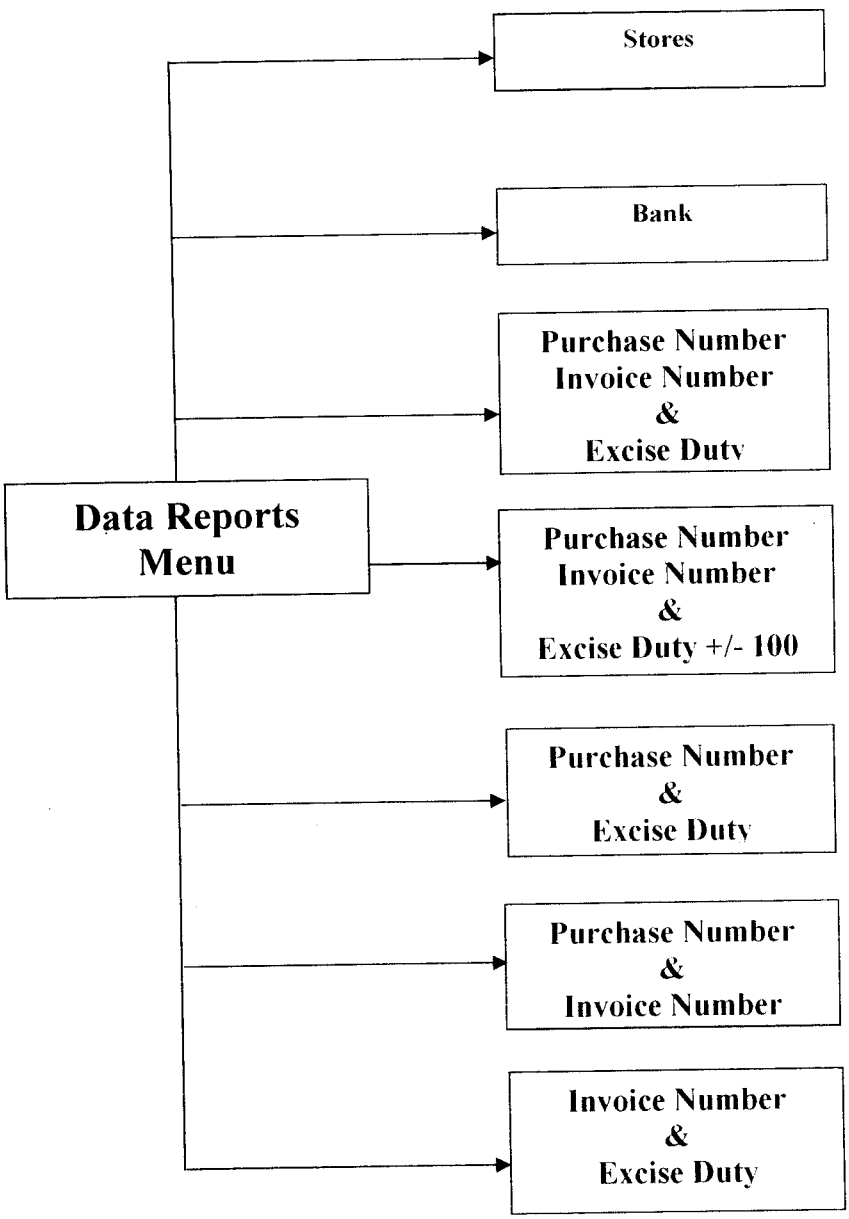
In table menu the actions like viewing table contents, deleting the entire table contents and to write SQL queries to perform operations on a table are provided. The text files obtained from bank and stores are appended to bank and stores tables respectively.

Reconciliation Form



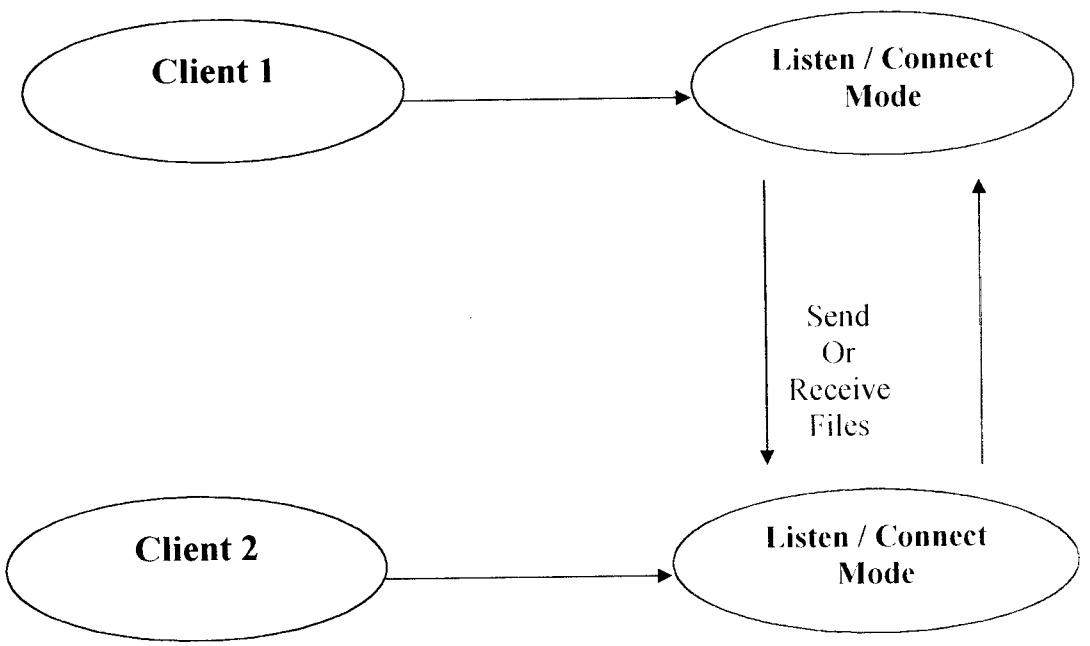
In this menu only both the bank and stores table's records are compared using six different types of matching. Records obtained from first five matching are displayed in data grids control and stored in separate tables to perform any changes. In unmatched case the unmatched records of bank stores are displayed in separate data grid controls and further modifications are done manually.

Reports Form



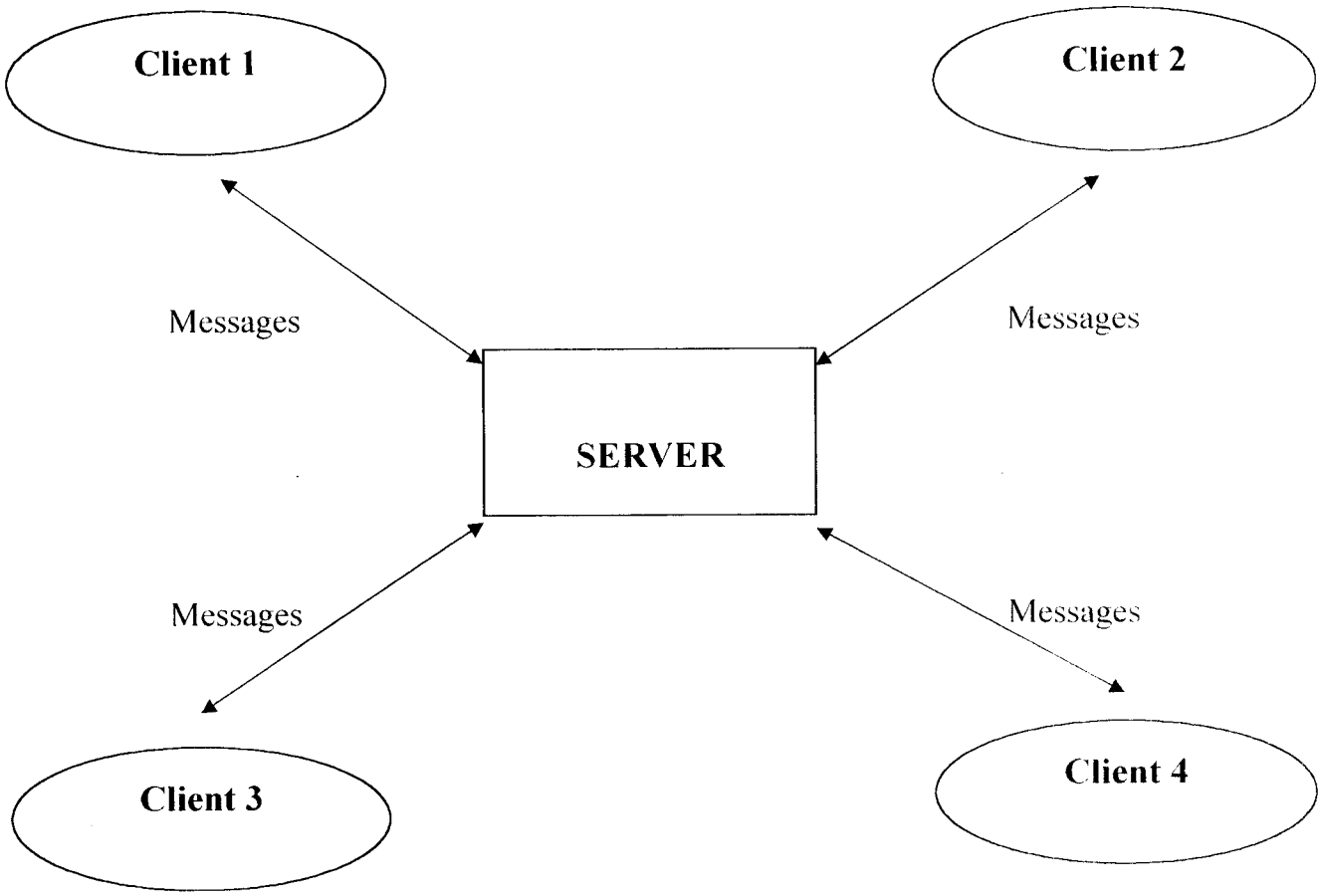
This menu provides reports for all the matching performed along with the calculated sum for Basic excise Duty column. These reports can be exported as HTML and other text files.

File Transfer



In file transfer application one of the clients should be in listening mode waiting for the connection. When another client sends a connection request to it, it accepts connection and both the clients can the transfer text files between them.

CHAT



In chat application the server should be started first which waits for connections from clients. All the clients have to first connect to the server and can then chat with other clients. All the messages will be routed through server only. The client can send message to all the available clients or to particular clients by getting clients list from server.

5. System Specification

5.1 Hardware Configuration

Operating System	:	Microsoft Windows 9x / NT
Processor	:	Pentium II
Memory	:	128 MB RAM
Hard Disk	:	10 GB
Monitor	:	14" Samsung Color Monitor
Cache	:	256 KB
Floppy Disk Drive	:	1.44 MB
CD ROM	:	52X
Key Board	:	104 keys

5.2 Software Configuration

Front-End : Visual Basic 6.0

Back-End : MS-Access

6. SYSTEM IMPLEMENTATION AND MAINTENANACE

Implementation is the stage, which is crucial in the life cycle of the new system designed. The main stage in the implementation is planning, training, system testing. Implementation means converting a new or revised system into an operational one. Conversion is the main aspect of implementation. It is the process of changing from the old system to the new one. After system is implemented, user conducts a review of the system. It is used to gather information for the maintenance of the system. The basic review methods is a data collection method of questionnaire, interview etc.

POST IMPLEMENTATION REVIEW:

The post implementation review is sometimes called system audit. The review is intended to accomplish two goals:

- Evaluate the operational information system that users developed.
- Evaluate the system development procedures to determine how the project could have been improved

MAINTENANCE:

Maintenance is one important phase in implementation. Maintenance by describing four activities that are under taken after a program is released for use.

The first maintenance activity occurs because it is unreasonable to assume that software testing will uncover all latent errors in a large software system. The process that includes the diagnosis and corrections of one or more error is called **corrective maintenance.**

Adaptive maintenance is an activity that modifies software to properly interface with the changing environment is both necessary and common.

The third activity is **perfective maintenance**, this activity accounts for the majority of all efforts expend on software maintenance.

The fourth maintenance activity occurs when software is changed to improve future maintainability or reliability or to provide a better basis for future enhancement often called **preventive maintenance** this activity characterized by reverse engineering and re-engineering.

Task association with software maintenance begins long before a request for maintenance is made. Initially a maintenance organization must be established, reporting and evaluation procedures must be described and a standardized sequence of event must be defined for each maintenance request.

In addition record-keeping procedures for maintenance activities should be established and review and evaluation criteria defined.

7. Conclusion

The system is designed in such an extendable fashion to incorporate the future changes in to the system easily. The various user-friendly features are introduced in this project. The system is developed according to the requirements produced by the organization. Needed documents are generated and adequate documentation has been provided for maintenance and further enhancements. The operation requirement by the user to operate the system is basic computer knowledge only.

8. Scope for Future Development

In the reconciliation process the unmatched records are handled manually due to various reasons which can be studied and improved to be performed by the system itself. In file transfer, only text files are transferred. It can be further enhanced to transfer any kind of files. Files can be encrypted and decrypted to ensure security and compressed during transmission to reduce the network traffic. Chat can be modified as voice chat and convert it into a video conferencing system.

9. Coding

Deletion Form

```
Option Explicit
Private Sub cancel_Click()
Unload delform
End Sub
Private Sub Form_Load()
   delform.tabcombo.Text = "match1"
   delform.tabcombo.AddItem "stores"
   delform.tabcombo.AddItem "bank"
   delform.tabcombo.AddItem "match1"
   delform.tabcombo.AddItem "match2"
   delform.tabcombo.AddItem "match3"
   delform.tabcombo.AddItem "match4"
   delform.tabcombo.AddItem "match5"
   delform.tabcombo.AddItem "unmatch"
   End Sub
Private Sub ok_Click()
   On Error GoTo errhandle
   Dim cmd As ADODB.Command
   Dim cn As ADODB.Connection
   Dim rs As ADODB.Recordset
   Dim sql As String, s As String
   Dim size As Integer
   Dim cmk As String
   cmk = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & datasrc &
";Persist Security Info=False;Jet OLEDB:Database Password=" & passwd
   Set cn = New ADODB.Connection
   With cn
```

```
.ConnectionString = cmk
.Open
End With
sql = "select * from " & delform.tabcombo.Text
Set rs = New ADODB.Recordset
rs.Open sql, cn, adOpenKeyset, adLockPessimistic
size = rs.RecordCount
If size <= 0 Then
s = MsgBox("No Records", vbInformation + vbOKOnly, "Information")
Unload delform
Exit Sub
Else
Set cmd = New ADODB.Command
Dim s1 As String
s1 = MsgBox("Sure to Delete", vbExclamation + vbYesNo, "Information")
If s1 = vbYes Then
cmd.CommandType = adCmdText
Set cmd.ActiveConnection = cn
cmd.CommandText = "delete from " & delform.tabcombo.Text
cmd.execute
cn.Close
' rs.Close
s1 = MsgBox(Str$(size) & " --> Records Deleted", , "Deletion")
Unload delform
Else
delform.Visible = True
End If
End If
```

View Form

```
Private Sub cancel_Click()
   Unload viewform
   SelectTable = False
   Unload Perfect_Match
End Sub
Private Sub Form_Load()
   viewcombo.Font.Name = "garamond"
   viewcombo.Text = "STORES"
   viewcombo.AddItem "STORES"
   viewcombo.AddItem "BANK"
   viewcombo.AddItem "MATCH1"
   viewcombo.AddItem "MATCH2"
   viewcombo.AddItem "MATCH3"
   viewcombo.AddItem "MATCH4"
   viewcombo.AddItem "MATCH5"
   viewcombo.AddItem "UNMATCH"
   SelectTable = True
End Sub
Private Sub ok_Click()
   On Error GoTo errhandle
   Dim cn As ADODB.Connection
   Dim rs As ADODB.Recordset
   Dim cmk$, s$
   cmk = "Provider=Microsoft.Jet.OLEDB.4.0;Data Source=" & datasrc &
";Persist Security Info=False;Jet OLEDB:Database Password=" & passwd
   Set cn = New ADODB.Connection
   With cn
```

```
.ConnectionString = cmk
.Open
End With
Set rs = New ADODB.Recordset
cn.CursorLocation = adUseClient
If LCase(viewcombo.Text) = "bank" Then
 Set rs = cn.execute("select,sec,das,ach,dinvno,dpono,dbed,cbed,dmonth,ind
          from bank")
ElseIf LCase(viewcombo.Text) = "stores" Then
Set rs = cn.execute("select
slno,date,invno,invdate,qty,cenvat,bed,pono,classfn,firmname,descpn,dbno,dbd
ate,ward,asse_code,doc,sup,ind,value,diff from stores")
Else
Set rs = cn.execute(viewcombo.Text)
End If
If rs.RecordCount <= 0 Then
Unload Me
MsgBox ("No Records")
Else
rs.MoveFirst
Call call2
Set Perfect_Match.DataGrid2.DataSource = rs
Perfect_Match.DataGrid2.Caption = viewcombo.Text
Perfect_Match.StatusBar1.Panels(3).Text = rs.RecordCount & " Records"
Perfect_Match.DataGrid2.Refresh
Unload Me
End If
```

Query form

```
Option Explicit
Private Sub cancel_Click()
Unload queryform
End Sub
Private Sub clear_Click()
querytext.Text = ""
End Sub
Private Sub execute_Click()
   Dim cn As ADODB.Connection
   Dim rs As ADODB.Recordset
   Dim cmk$, s$, res$, Tablename$
   'On Error GoTo errhandle
   cmk = "Provider=Microsoft.Jet.OLEDB.4.0:Data Source=" & datasrc &
";Persist Security Info=False;Jet OLEDB:Database Password=" & passwd
   Set cn = New ADODB.Connection
   With cn
   .ConnectionString = cmk
   .Open
   End With
   Set rs = New ADODB.Recordset
   cn.CursorLocation = adUseClient
   res = LCase(Trim$(querytext.Text))
   If InStr(res, "select") Then
      Set rs = cn.execute(querytext.Text)
   Else
      cn.execute (querytext.Text)
      If InStr(res, "bank") Then Tablename = "bank"
```

```
        If InStr(res, "stores") Then Tablename = "stores"
        If InStr(res, "match1") Then Tablename = "match1"
        If InStr(res, "match2") Then Tablename = "match2"
        If InStr(res, "match3") Then Tablename = "match3"
        If InStr(res, "match4") Then Tablename = "match4"
        If InStr(res, "match5") Then Tablename = "match5"
        If InStr(res, "unmatch") Then Tablename = "unmatch"
            rs.Open "select * from " & Tablename, cn
    End If
    rs.MoveFirst
    If rs.RecordCount = 0 Then
    Unload Me
    MsgBox ("No Records")
    Else
    Call call2
    Set Perfect_Match.DataGrid2.DataSource = rs
    Perfect_Match.StatusBar1.Panels(3).Text = rs.RecordCount & " Records"
    Perfect_Match.DataGrid2.Refresh
    End If
errhandle:
    If Err Then
    s = MsgBox(Err.Description, , "Error Message")
    End If
End Sub
Private Sub Form_Activate()
querytext.Text = ""
End Sub
```

File Transfer

```
Option Explicit
Dim nameoffile$, filepath$, contents$, dirname$, s$
Dim filenum%, pos%
Dim isopen As Boolean
Dim portn%
Private Sub cmdSend_Click()
   Call tcpclient.SendData(contents)
End Sub
Private Sub Command1_Click()
RichTextBox1.Text = ""
End Sub
Private Sub connect_click()
'On Error GoTo errhandle
   tcpclient.RemoteHost = _
   InputBox("Enter the remote host IP address or remote host Name" & vbCrLf
& vbCrLf & vbTab & "( 192.168.12.08  or  System-1 )", _
         "IP Address", "192.168.0.1")
      If tcpclient.RemoteHost = "" Then
      tcpclient.RemoteHost = "192.168.0.1"
   End If
   tcpclient.RemotePort = 5000
   Call tcpclient.connect
errhandle:
   If err Then
   MsgBox (err.Description)
   tcpclient.Close
   tcpclient.LocalPort = 5000
```

```
    Call tcpclient.Listen
    End If
End Sub
Private Sub Form_Load()
RichTextBox1.Text = ""
txtoutput.Text = ""
End Sub
Private Sub Form_Terminate()
    End
End Sub
Private Sub quit_Click()
End
End Sub
Private Sub tcpClient_Close()
    cmdsend.Enabled = False
    connect.Enabled = True
    Call tcpclient.Close
    txtoutput.Text = _
        txtoutput.Text & "Remote Host closed connection." & vbCrLf & vbCrLf
    txtoutput.SelStart = Len(txtoutput.Text)
    tcpclient.LocalPort = 5000
    tcpclient.Listen
End Sub
Private Sub tcpClient_Connect()
    txtoutput.Text = "*** Connected to IP Address:" & tcpclient.RemoteHostIP
& " . Port #:" & _
        tcpclient.RemotePort & vbCrLf & vbCrLf
        connect.Enabled = False
        cmdsend.Enabled = True
```

```
End Sub

Private Sub tcpClient_ConnectionRequest(ByVal requestID As Long)
   If tcpclient.State <> sckClosed Then
      Call tcpclient.Close
   End If
   tcpclient.Accept (requestID)
   txtoutput = txtoutput.Text & "*** " & _
      "Request From IP:" & tcpclient.RemoteHostIP & _
      ". Remote Port: " & tcpclient.RemotePort & vbCrLf & vbCrLf
   cmdsend.Enabled = True
   save.SetFocus
End Sub
Private Sub tcpClient_DataArrival(ByVal bytesTotal As Long)
   Dim message As String
   Call tcpclient.GetData(message)
   RichTextBox1.Text = RichTextBox1.Text & vbCrLf & message
End Sub
Private Sub tcpClient_Error(ByVal Number As Integer, Description As String,
ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As String,
ByVal HelpContext As Long, CancelDisplay As Boolean)
   Dim result As Integer
   If Number = 10061 Then
      txtoutput.Text = "Cannot Connect to RemoteHost" & vbCrLf & vbCrLf
      txtoutput.Text = txtoutput.Text & "Wait for an incoming connection....."
& vbCrLf & vbCrLf
   Else
      result = MsgBox(Source & ": " & Description, _
         vbOKOnly, "TCP/IP Error")
```

```
    End If
    tcpclient.Close
    tcpclient.LocalPort = 5000
    Call tcpclient.Listen
    save.SetFocus
End Sub
Private Sub browse_Click()
On Error GoTo errhandle
open1.ShowOpen
file1.Text = open1.FileName
filepath = open1.FileName
Call ReadContents
errhandle:
End Sub
Public Sub ReadContents()
 On Error GoTo errhandle
 filenum = FreeFile()
 Open filepath For Input As filenum
 isopen = True
 contents = Input$(LOF(filenum), filenum)
 cmdsend.Enabled = True
errhandle:
If err Then
MsgBox (err.Description)
End If
End Sub
Private Sub save_Click()
On Error GoTo err
Dim name$
```

```
 savedialog.Flags = cdlOFNOverwritePrompt
 savedialog.ShowSave
 If savedialog.FileName <> "" Then
   name = savedialog.FileName
   filenum = FreeFile()
   Open name For Output As #filenum
   Print #filenum, RichTextBox1.Text
   Close #filenum
   RichTextBox1.Text = ""
   Exit Sub
 End If
err:
End Sub
Private Sub tcpclient_SendComplete()
   MsgBox ("File Successfully Sent")
End Sub
```

Chat-Server

```
Option Explicit
Const SERVER_LIST = "[ClientList]"
Const SCK_CODE_ALL = "[All]"
Const SCK_CODE_CLIENT = "[Client]"
Const SCK_CODE_IP = "{Ip}"
Const I_WANT_LIST = "[INeed]"
Public MAXCONNECTED As Integer
Public freesock As Long
Private ServerIndex As Long
Public mesg$
Dim i%
Private Sub closeserver_Click()
   sockserver(0).Close
   End
End Sub
Private Sub Form_Load()
   ServerIndex = 0
   sockserver(0).Protocol = sckTCPProtocol
   sockserver(0).LocalPort = 5000
   sockserver(0).Listen
   Label2.Caption = Label2.Caption & " " & sockserver(0).LocalHostName
   Label3.Caption = Label3.Caption & " " & sockserver(0).LocalIP
End Sub
Private Sub sockserver_Close(Index As Integer)
 Dim clientname As String
 sockserver(Index).Close
   For i = 0 To List1.ListCount - 1
```

```
      clientname = Left(List1.List(i), InStr(List1.List(i), "->") - 1)
    If Val(clientname) = Index Then
      List1.RemoveItem (i)
      Exit Sub
    End If
   Next i
End Sub
Private Sub sockserver_ConnectionRequest(Index As Integer, ByVal
requestID As Long)
 Dim i As Long, remoteip$, remotename$
 freesock = 0
   For i = 1 To ServerIndex
      If sockserver(i).State = sckClosed Then
      freesock = i
      MAXCONNECTED = MAXCONNECTED - 1
      Exit For
      End If
   Next i
   If freesock = 0 Then
    ServerIndex = ServerIndex + 1
    Load sockserver(ServerIndex)
    freesock = ServerIndex
   End If
    sockserver(freesock).LocalPort = 0
    sockserver(freesock).Accept requestID
    remoteip = sockserver(freesock).RemoteHostIP
    remotename = sockserver(freesock).RemoteHost
    List1.AddItem freesock & "->" & remoteip & remotename
    MAXCONNECTED = MAXCONNECTED + 1
```

```
End Sub

Private Sub sockserver_DataArrival(Index As Integer, ByVal bytesTotal As
Long)
 Dim data As String, From_Me As String
 Dim i%, ipindexvalue%
 Dim sCommand As String
 Dim sInstruction As String
 Dim ipname$, ipaddress$, ipindex$
 Dim sData As String
 Dim j%
data = String(bytesTotal + 2, Chr$(0))
 sockserver(Index).GetData data, vbString, bytesTotal
 From_Me = sockserver(Index).RemoteHostIP & ":"
 messages.Text = messages.Text & From_Me & data
Do While InStr(1, data, vbCrLf)
   sCommand = Mid(data, 1, InStr(1, data, vbCrLf) - 1)
   sInstruction = Mid(sCommand, 1, InStr(1, sCommand, "]"))
   sData = Mid(sCommand, InStr(1, sCommand, "]") + 1, Len(sCommand))
   Select Case sInstruction
      Case I_WANT_LIST
         mesg = ""
         For i = 0 To List1.ListCount - 1
            mesg = mesg + List1.List(i) & "%"
         Next
         mesg = SERVER_LIST & mesg & vbCrLf
         sockserver(Index).SendData mesg

      Case SCK_CODE_ALL
```

```
            For i = 1 To MAXCONNECTED
            If sockserver(i).State = sckConnected And i <> Index Then
              sockserver(i).SendData From_Me & " >> " & sData
              Call WasteTime(1)
            End If
           Next i
      Case SCK_CODE_CLIENT
            ipname = Mid(sData, 1, InStr(1, sData, "{") - 1)
            ipname = Trim(ipname)
            sData = Mid(sData, InStr(1, sData, "}") + 1, Len(sData))
            For i = 0 To List1.ListCount - 1
             ipaddress = Mid(List1.List(i), InStr(List1.List(i), "->") + 2, _
                         Len(List1.List(i)))
             ipindex = Left(List1.List(i), InStr(List1.List(i), "->") - 1)
             ipaddress = Trim(ipaddress)
             ipindexvalue = CInt(ipindex)
            If StrComp(ipname, ipaddress, vbTextCompare) = 0 Then
              If sockserver(ipindexvalue).State = sckConnected Then
                sockserver(ipindexvalue).SendData From_Me & " >> " & sData
              End If
              If sockserver(ipindexvalue).State = sckClosed Then
                sockserver(Index).SendData "From_Server --> Client is
Disconnected"
              End If
            End If
            Next i
   End Select
   data = Mid(data, InStr(1, data, vbCrLf) + 2, Len(data))
Loop
```

```
End Sub
```

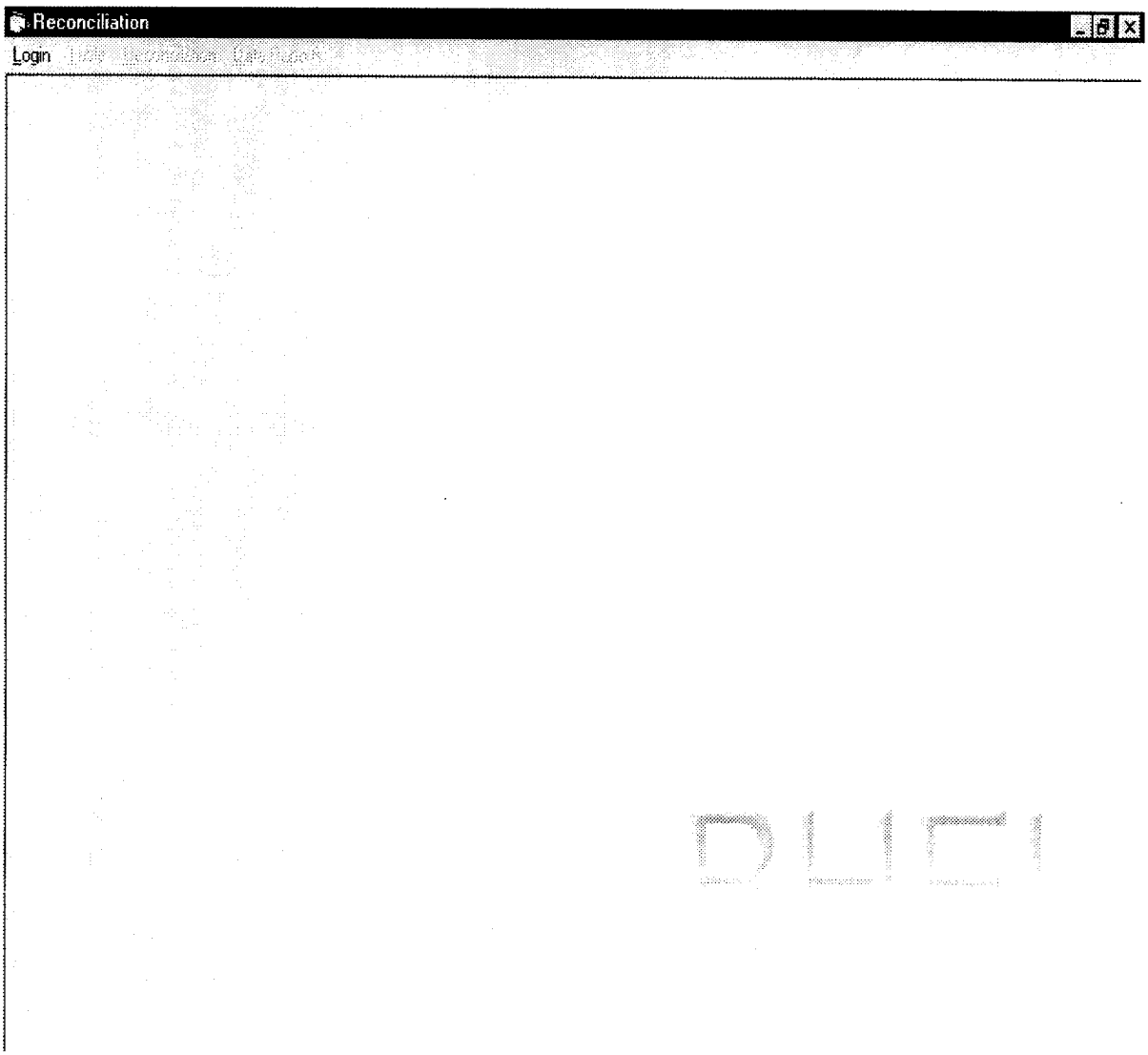
Chat – Client

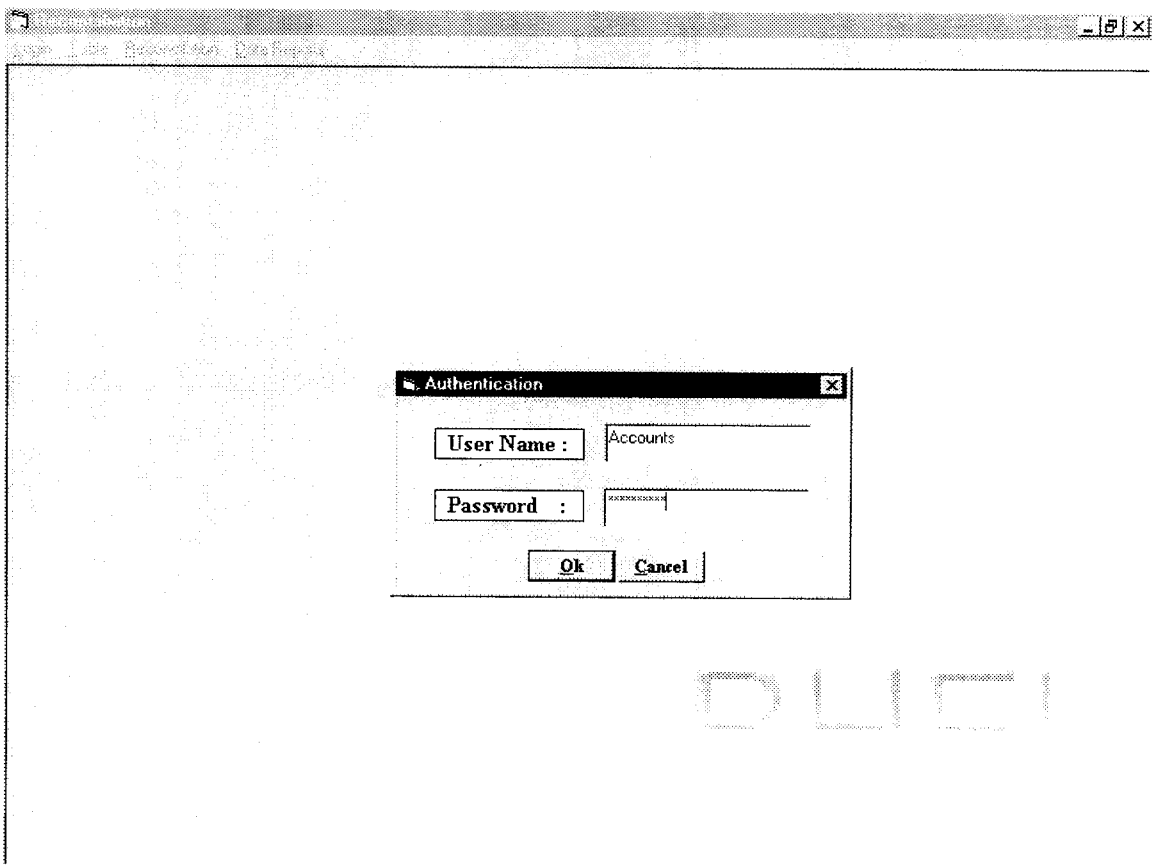
```
Option Explicit
Const SERVER_LIST = "[ClientList]"
Const SCK_CODE_ALL = "[All]"
Const SCK_CODE_CLIENT = "[Client]"
Const SCK_CODE_IP = "{Ip}"
Const I_WANT_LIST = "[INeed]"
Dim Server_Name As String
Dim data As String
Dim errmsg$
Private Sub all_Click()
   If sockclient.State = sckConnected Then
   sockclient.SendData SCK_CODE_ALL & mymessage.Text & vbCrLf
   End If
End Sub
Private Sub disconnect_Click()
   sockclient.Close
   End
End Sub
Private Sub Form_Load()
   sockclient.Protocol = sckTCPProtocol
   Server_Name = InputBox("Enter Server Name or Server IP Address", _
            "Server Connection", , 2000, 2000)
   If Server_Name = "" Then
   End
   End If
   Call Connect_to_Server
End Sub
```

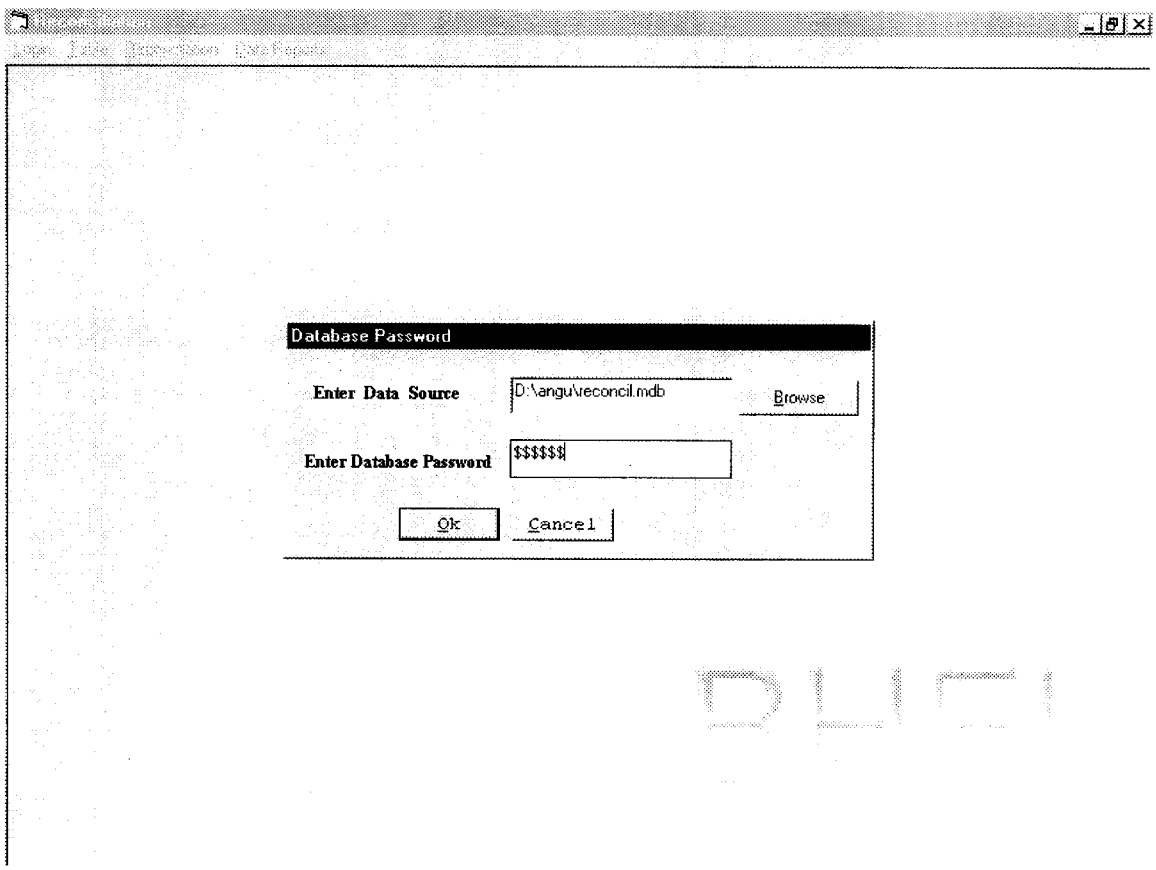
```
Private Sub sendata_Click()
   If sockclient.State = sckConnected Then
      sockclient.SendData  SCK_CODE_CLIENT  &  clientname.Text  &
SCK_CODE_IP & mymessage.Text & vbCrLf
   Else
      MsgBox "Unexpected Error"
      Call disconnect_Click
   End If
End Sub
Private Sub sockclient_Close()
   sockclient.Close
   End
End Sub
Private Sub sockclient_Connect()
   MsgBox "Connected to Server"
End Sub
Private Sub sockclient_DataArrival(ByVal bytesTotal As Long)
Dim i As Integer
Dim lstitem$
Dim sCommand As String
Dim sInstruction As String
Dim sData As String
data = ""
   data = String(bytesTotal + 2, Chr$(0))
   sockclient.GetData data, vbString, bytesTotal

Do While InStr(1, data, vbCrLf)
   sCommand = Mid(data, 1, InStr(1, data, vbCrLf) - 1)
```

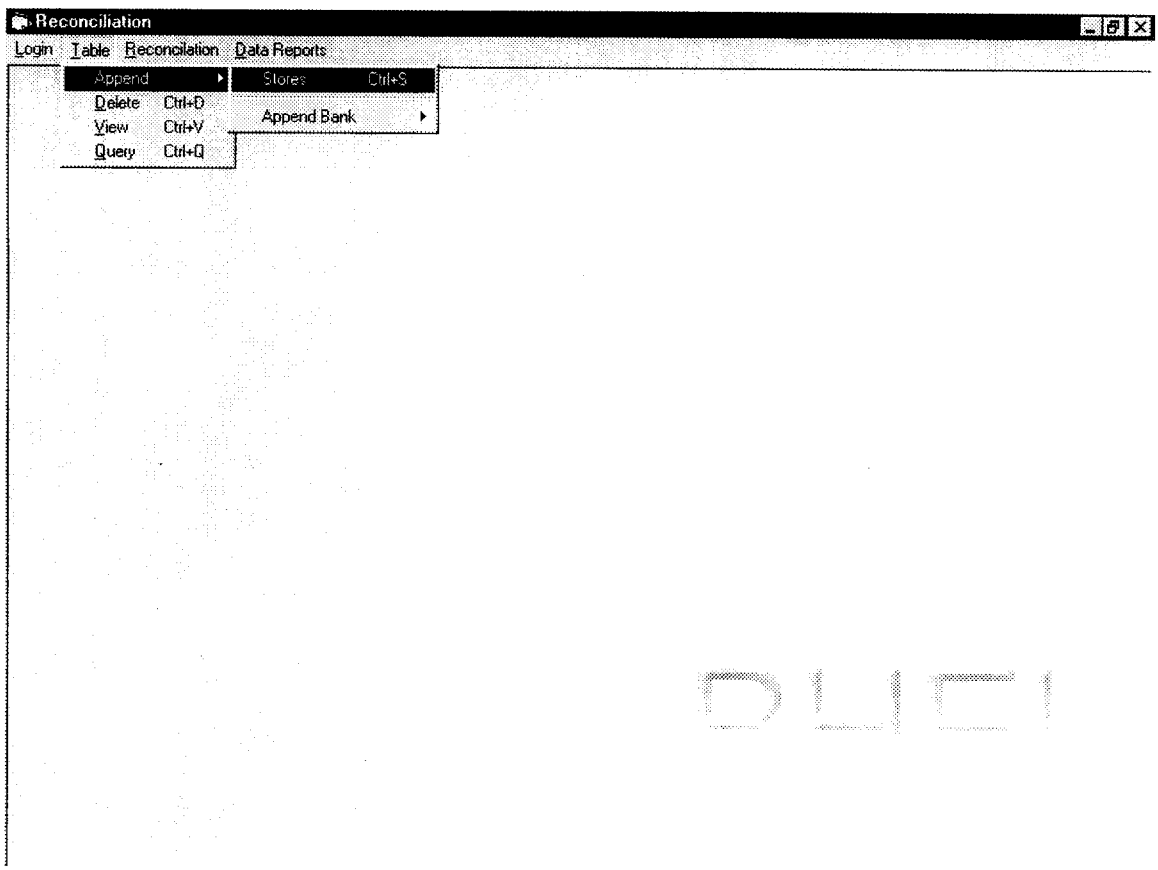
```
    sInstruction = Mid(sCommand, 1, InStr(1, sCommand, "]"))
    sData = Mid(sCommand, InStr(1, sCommand, "]") + 1, Len(sCommand))
    remotemessage.Text = remotemessage & data
        Select Case sInstruction
                Case SERVER_LIST
               recp.Clear
                Do While sData <> ""
                  lstitem = Mid(sData, 1, InStr(1, sData, "%") - 1)
                  sData = Mid(sData, InStr(1, sData, "%") + 1, Len(sData))
                  recp.AddItem lstitem
                Loop
          End Select
    data = Mid(data, InStr(1, data, vbCrLf) + 2, Len(data))
Loop
End Sub
Private Sub sockclient_Error(ByVal Number As Integer, Description As
String, ByVal Scode As Long, ByVal Source As String, ByVal HelpFile As
String, ByVal HelpContext As Long, CancelDisplay As Boolean)
    errmsg = MsgBox(Description & Err.Number, vbOKOnly, "Warning")
    Call sockclient.Close
    End
End Sub
Public Sub Connect_to_Server()
On Error GoTo errhandle
    sockclient.LocalPort = 0
    sockclient.RemoteHost = Server_Name
    sockclient.RemotePort = 5000
    Call sockclient.Connect
errhandle:
```

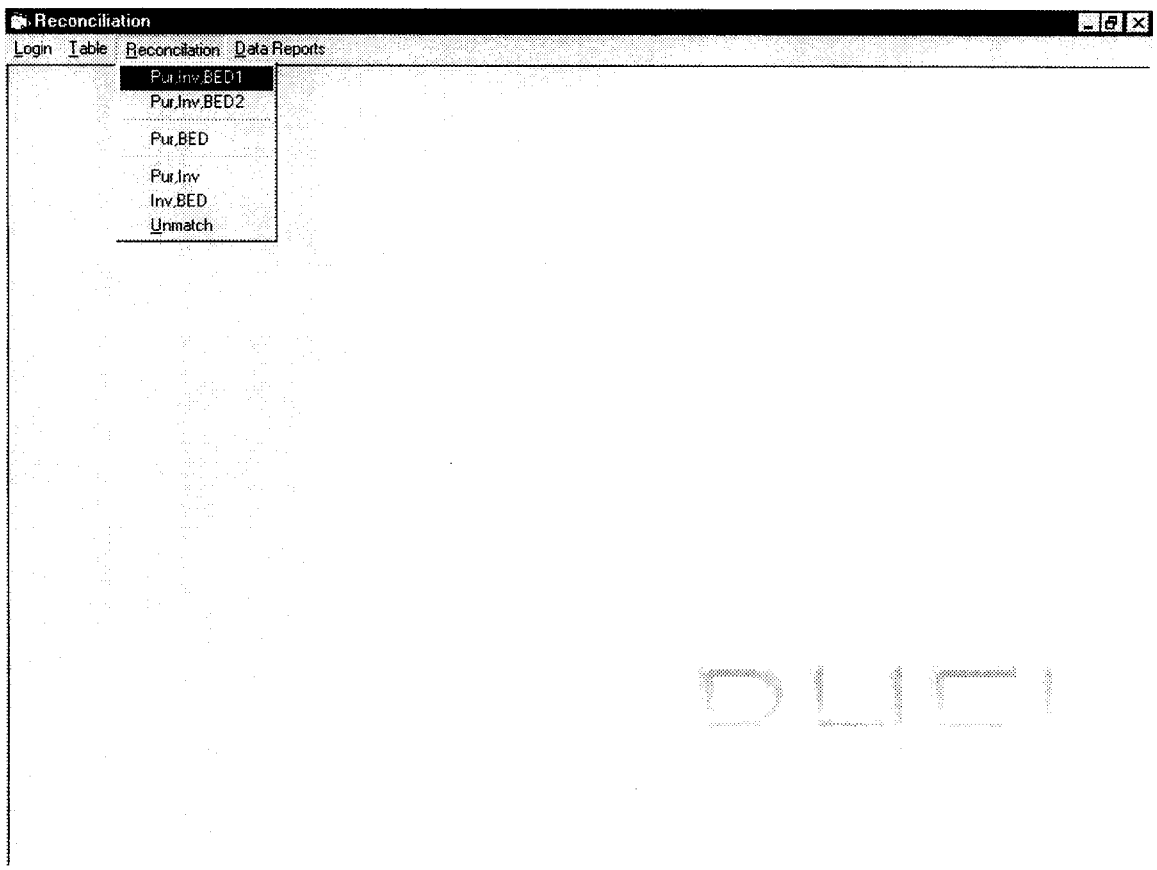
```
    If Err Then
    errmsg = MsgBox(Err.Description, vbOKOnly, "Warning")
    End
    End If
End Sub
Private Sub User_List_Click()
    If sockclient.State = sckConnected Then
        sockclient.SendData I_WANT_LIST & vbCrLf
    Else
        MsgBox "Unexpected Error"
        Call disconnect_Click
    End If
End Sub
```

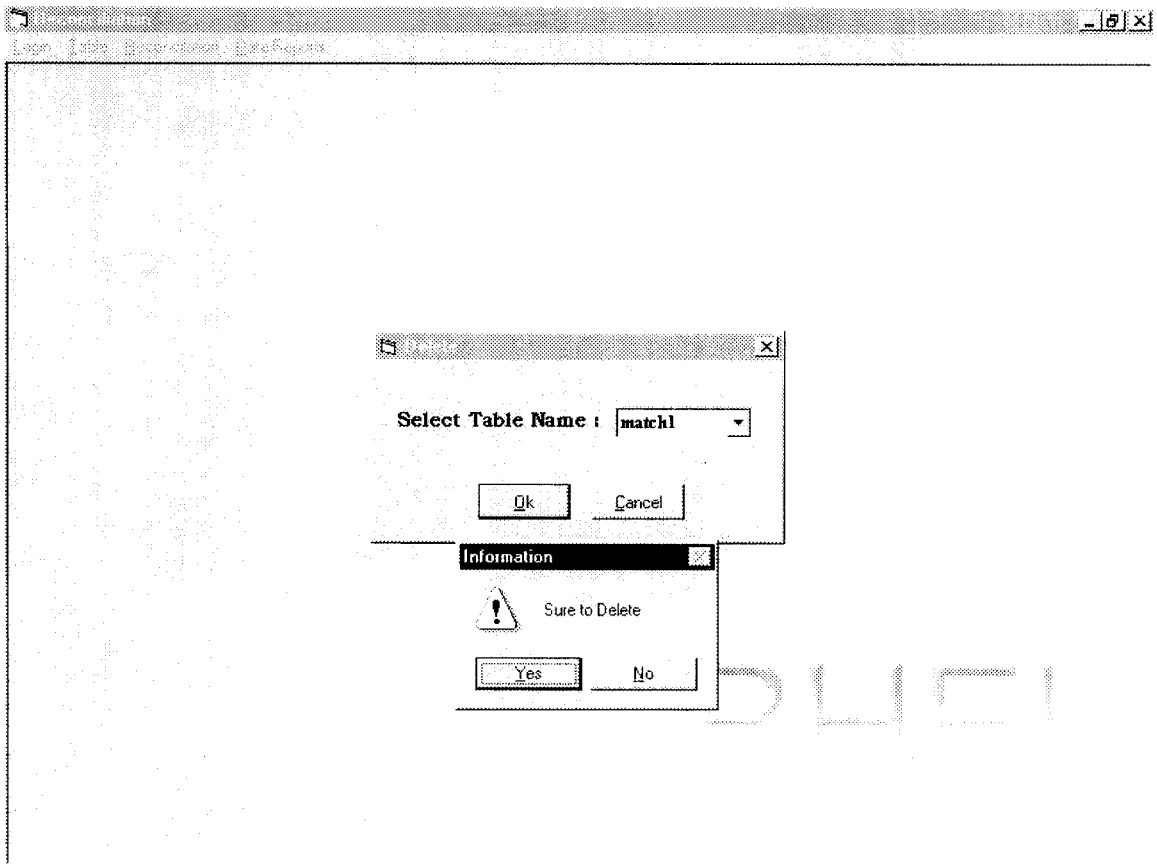

Reconciliation
Login

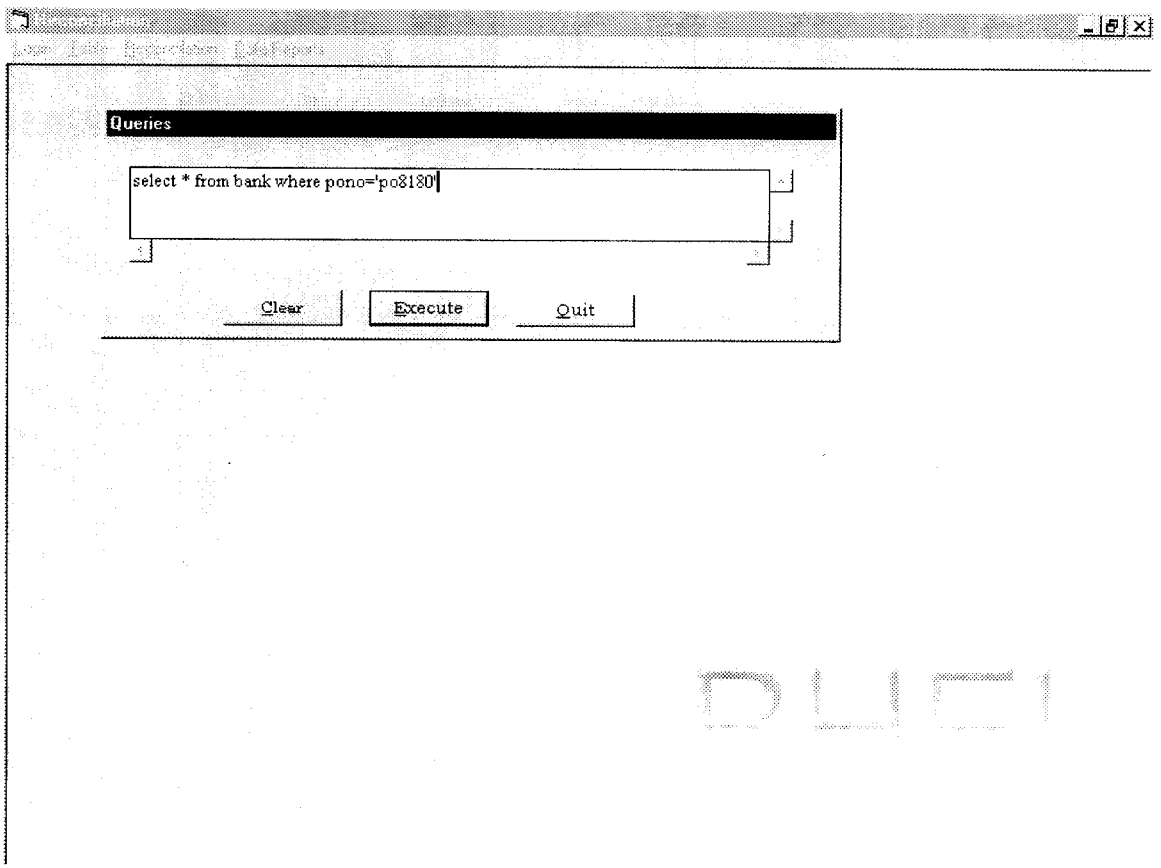

Authentication
User Name :
Accounts
Password :
Ok
Cancel

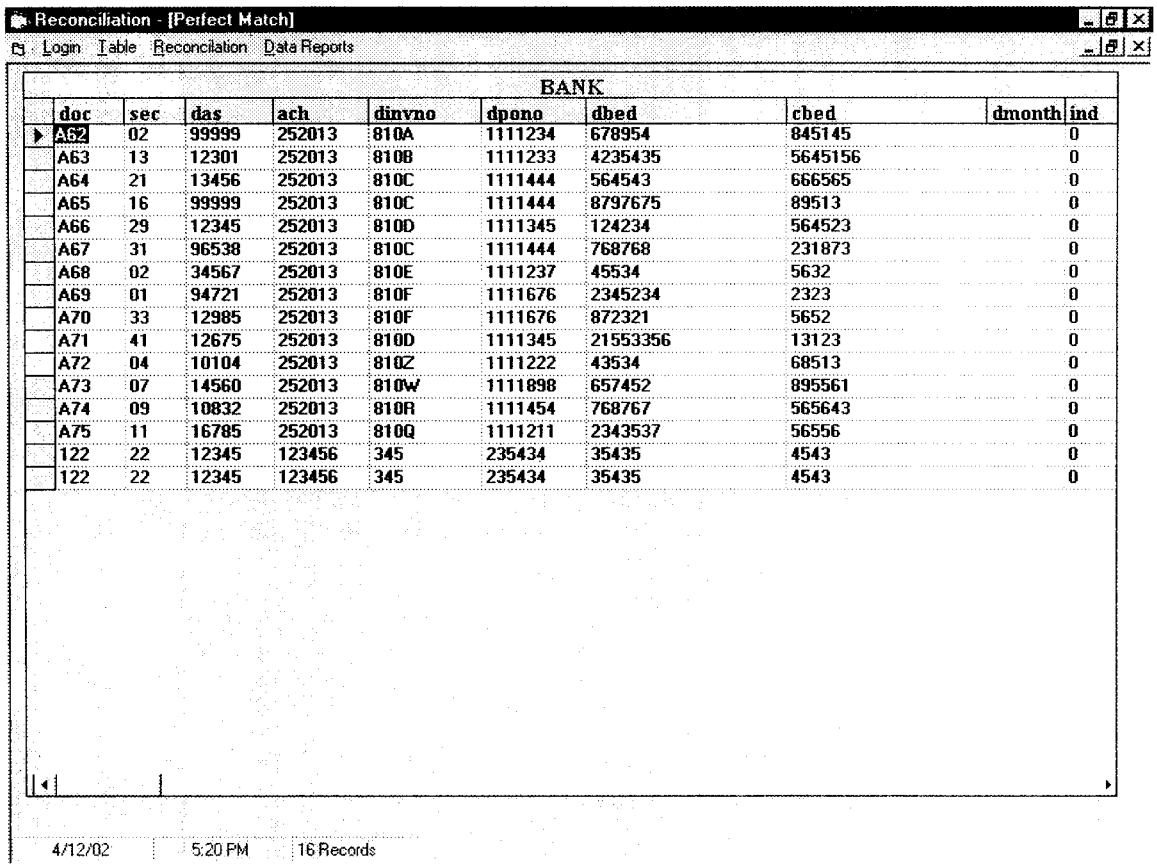

Database Password
Enter Data Source
D:\angu\reconcil.mdb
Browse
Enter Database Password
$$$$$$
Ok
Cancel

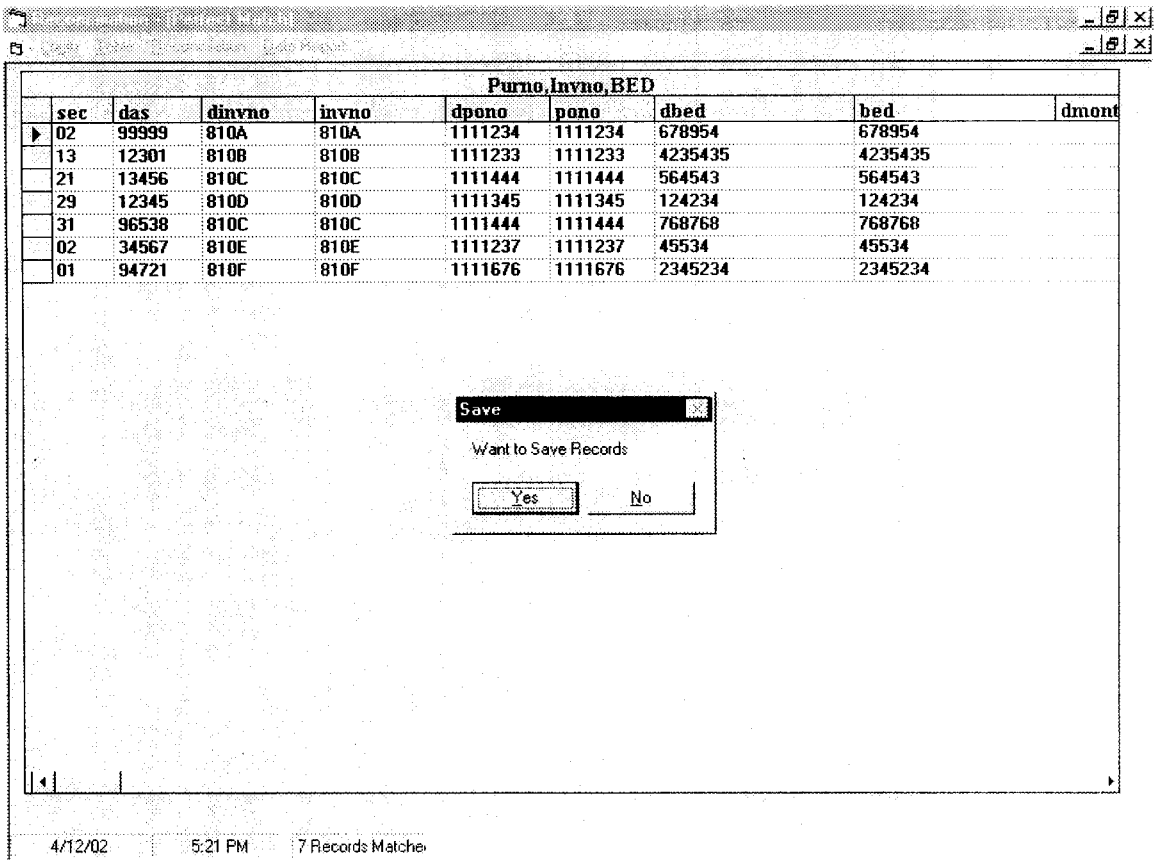

Reconciliation
Login
Table
Reconciliation
Data Reports
Append
Delete Ctrl+D
View Ctrl+V
Query Ctrl+Q
Stores Ctrl+S
Append Bank

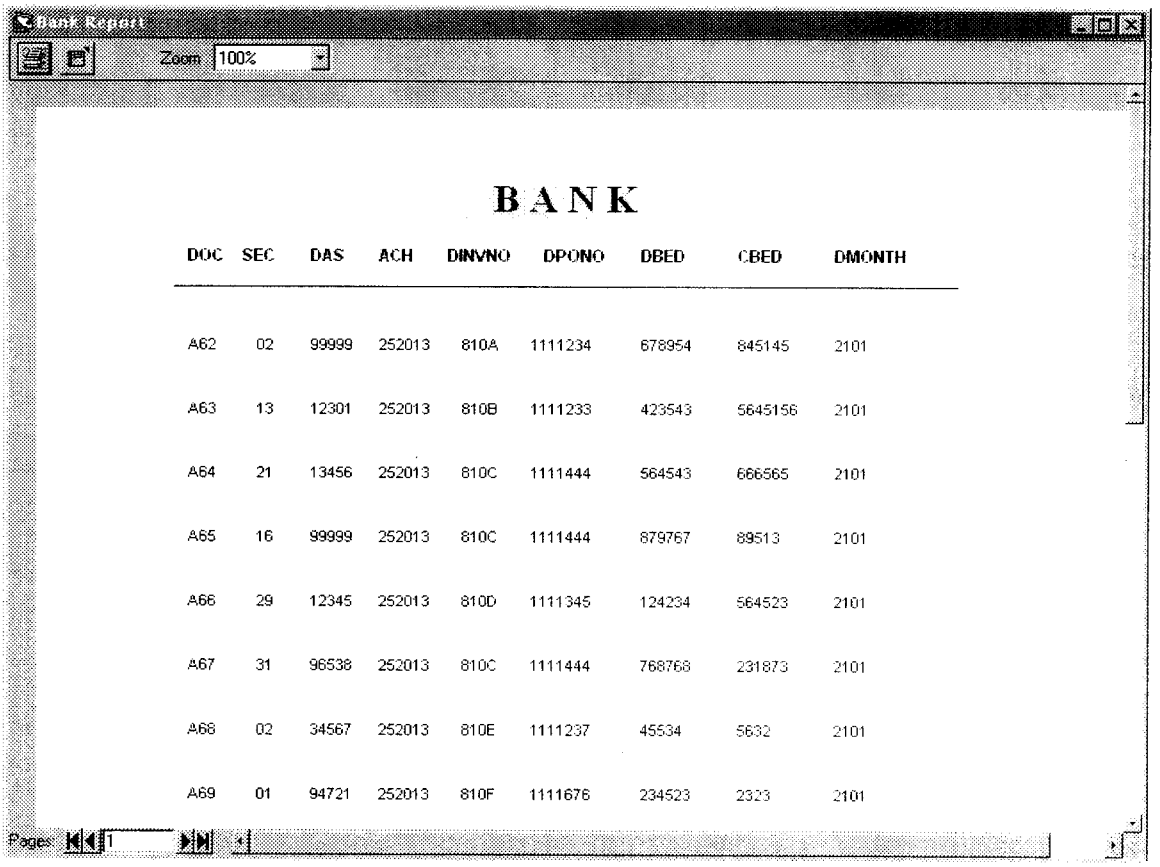

Reconciliation
Login
Table
Reconciliation
Data Reports
Pur,Inv,BED1
Pur,Inv,BED2
Pur,BED
Pur,Inv
Inv,BED
Unmatch

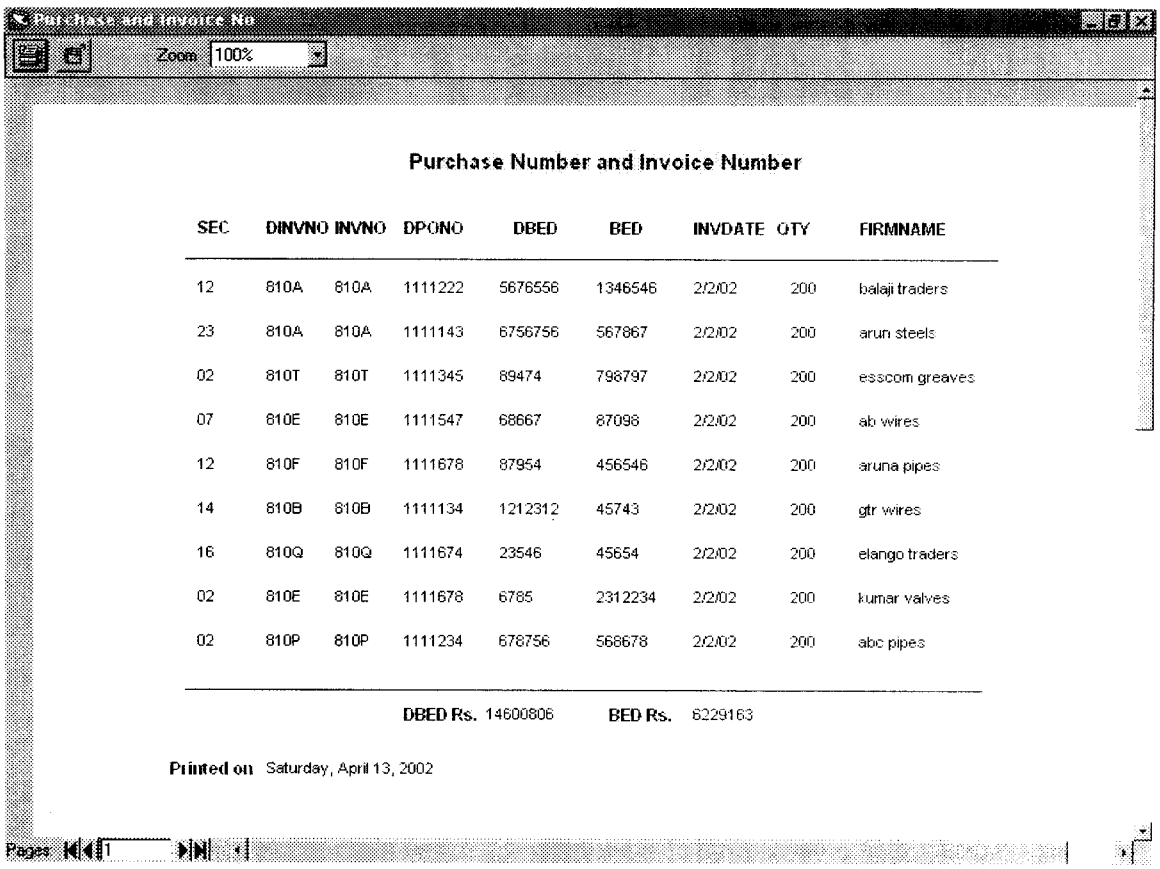

Select Table Name :
match1
Ok
Cancel
Information
Sure to Delete
Yes
No


Queries
select * from bank where pono='po8180'
Clear
Execute
Quit


Reconciliation - [Perfect Match]
Login Table Reconciliation Data Reports
BANK
doc sec das ach dinvno dpono dbed cbed dmonth ind
A62 02 99999 252013 810A 1111234 678954 845145 0
A63 13 12301 252013 810B 1111233 4235435 5645156 0
A64 21 13456 252013 810C 1111444 564543 666565 0
A65 16 99999 252013 810C 1111444 8797675 89513 0
A66 29 12345 252013 810D 1111345 124234 564523 0
A67 31 96538 252013 810C 1111444 768768 231873 0
A68 02 34567 252013 810E 1111237 45534 5632 0
A69 01 94721 252013 810F 1111676 2345234 2323 0
A70 33 12985 252013 810F 1111676 872321 5652 0
A71 41 12675 252013 810D 1111345 21553356 13123 0
A72 04 10104 252013 810Z 1111222 43534 68513 0
A73 07 14560 252013 810W 1111898 657452 895561 0
A74 09 10832 252013 810R 1111454 768767 565643 0
A75 11 16785 252013 810Q 1111211 2343537 56556 0
122 22 12345 123456 345 235434 35435 4543 0
122 22 12345 123456 345 235434 35435 4543 0
4/12/02 5:20 PM 16 Records


Purno,Invno,BED
sec
das
dinvno
invno
dpono
pono
dbed
bed
dmont
02 99999 810A 810A 1111234 1111234 678954 678954
13 12301 810B 810B 1111233 1111233 4235435 4235435
21 13456 810C 810C 1111444 1111444 564543 564543
29 12345 810D 810D 1111345 1111345 124234 124234
31 96538 810C 810C 1111444 1111444 768768 768768
02 34567 810E 810E 1111237 1111237 45534 45534
01 94721 810F 810F 1111676 1111676 2345234 2345234
Save
Want to Save Records
Yes
No
4/12/02
5:21 PM
7 Records Matche


Bank Report
Zoom 100%
BANK
DOC SEC DAS ACH DINVNO DPONO DBED CBED DMONTH
A62 02 99999 252013 810A 1111234 678954 845145 2101
A63 13 12301 252013 810B 1111233 423543 5645156 2101
A64 21 13456 252013 810C 1111444 564543 666565 2101
A65 16 99999 252013 810C 1111444 879767 89513 2101
A66 29 12345 252013 810D 1111345 124234 564523 2101
A67 31 96538 252013 810C 1111444 768768 231873 2101
A68 02 34567 252013 810E 1111237 45534 5632 2101
A69 01 94721 252013 810F 1111678 234523 2323 2101
Page: 1

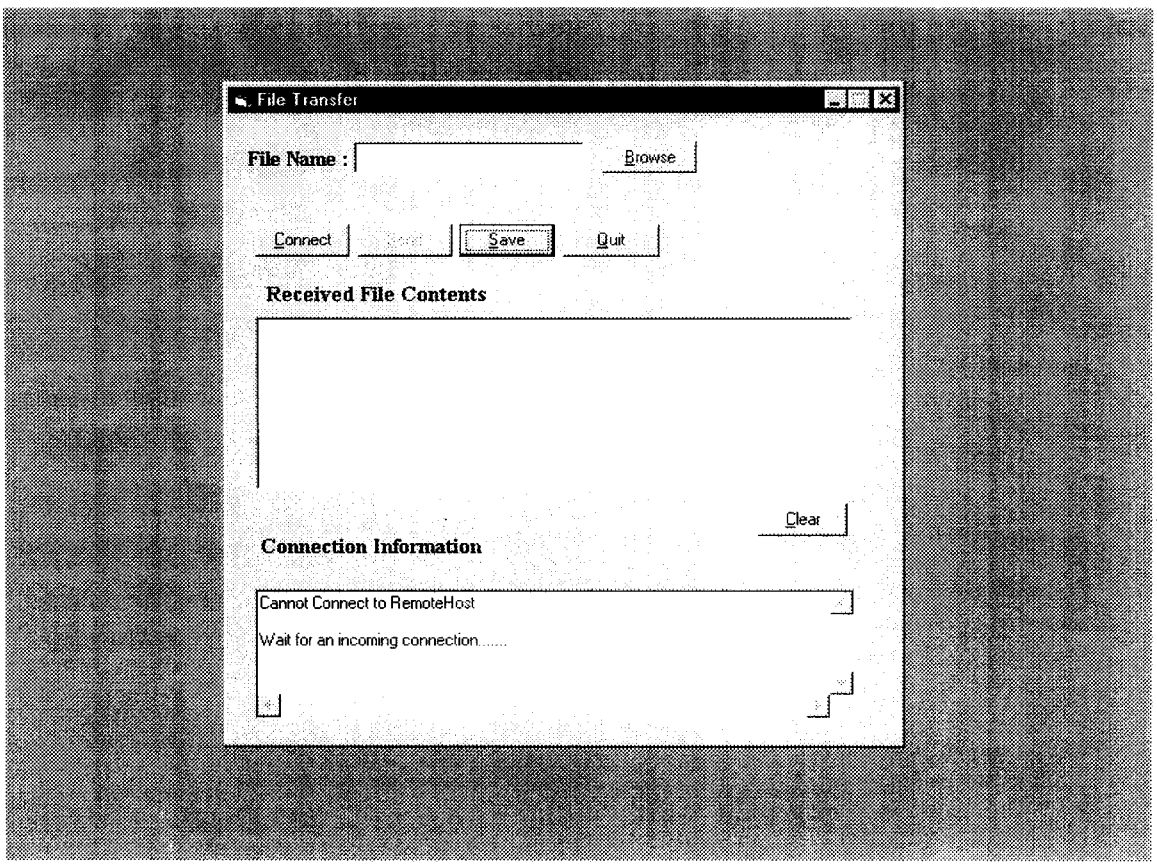


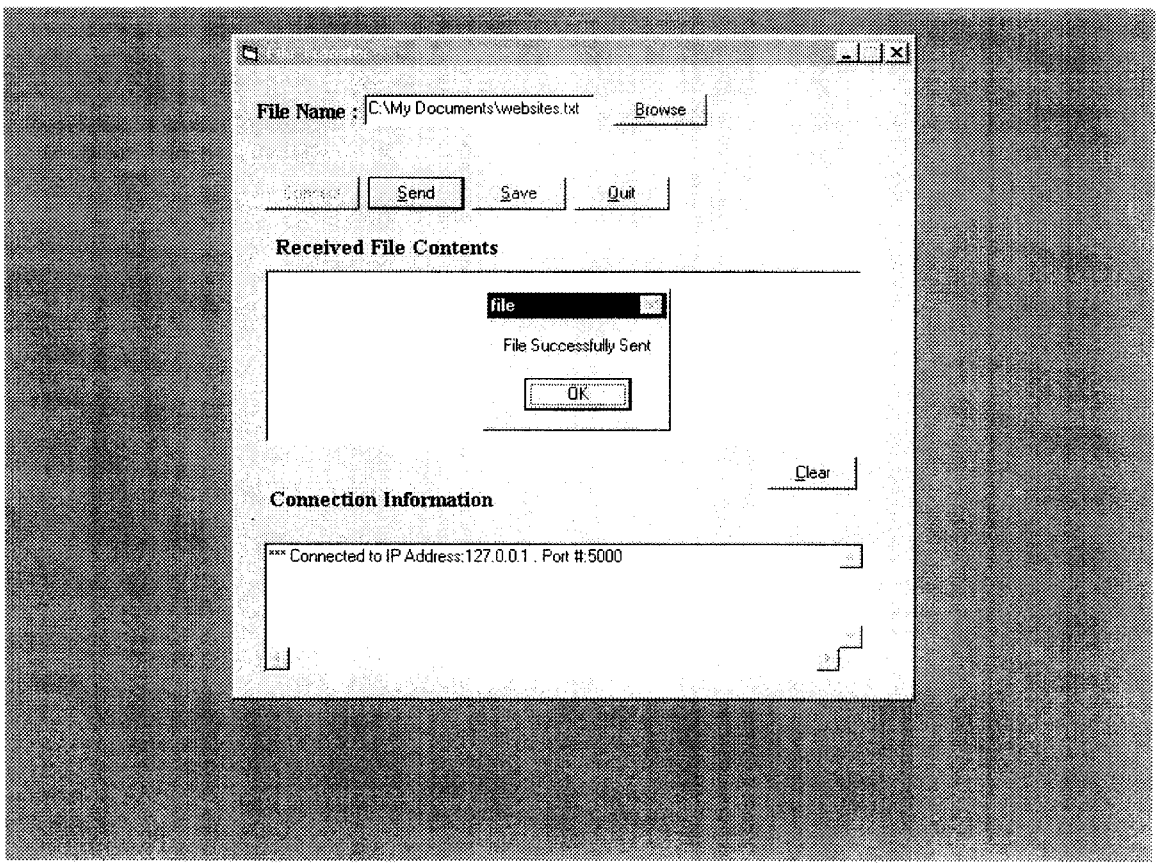
Purchase Number and Invoice Number

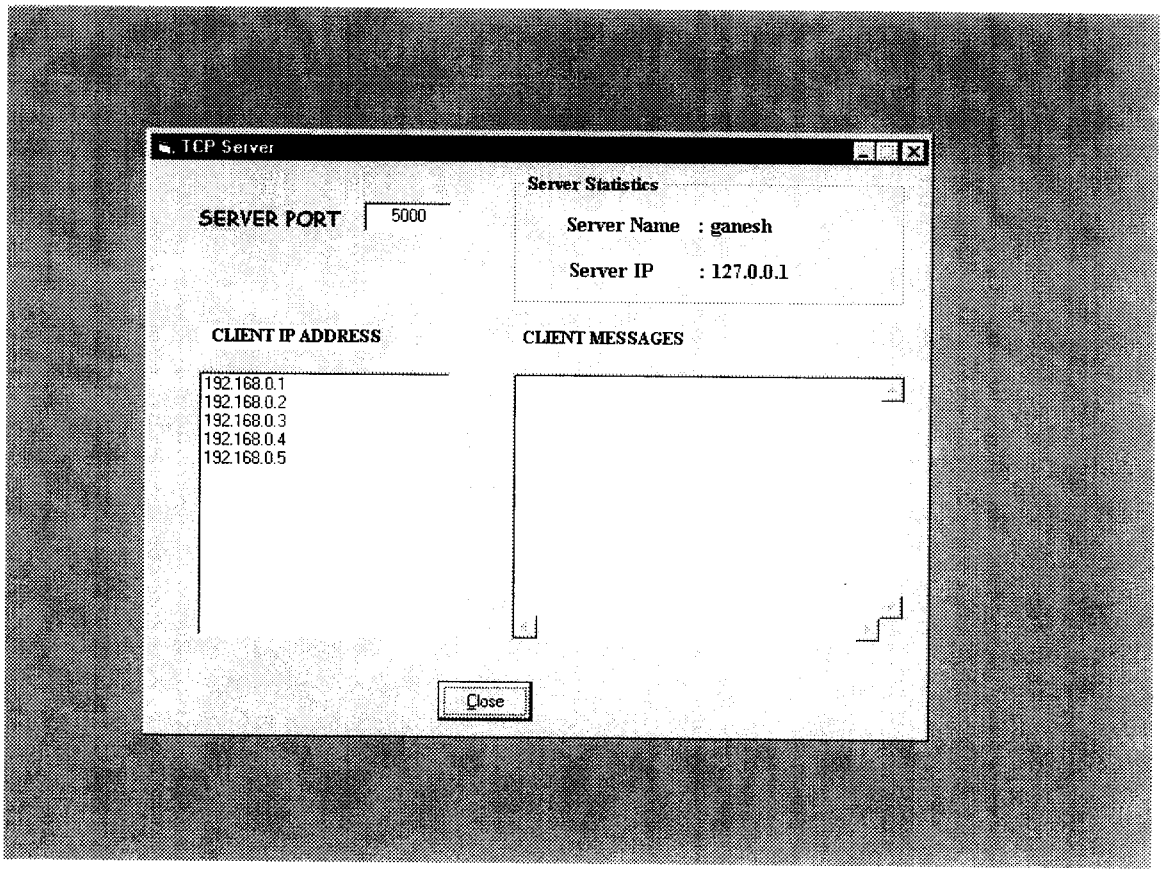
SEC	DINVNO	INVNO	DPONO	DBED	BED	INVDATE	QTY	FIRMNAME
12	810A	810A	1111222	5676556	1346546	2/2/02	200	balaji traders
23	810A	810A	1111143	6756756	567867	2/2/02	200	arun steels
02	810T	810T	1111345	89474	798797	2/2/02	200	esscom greaves
07	810E	810E	1111547	68667	87098	2/2/02	200	ab wires
12	810F	810F	1111678	87954	456546	2/2/02	200	aruna pipes
14	810B	810B	1111134	1212312	45743	2/2/02	200	gtr wires
16	810Q	810Q	1111674	23546	45654	2/2/02	200	elango traders
02	810E	810E	1111678	6785	2312234	2/2/02	200	kumar valves
02	810P	810P	1111234	678756	568678	2/2/02	200	abc pipes

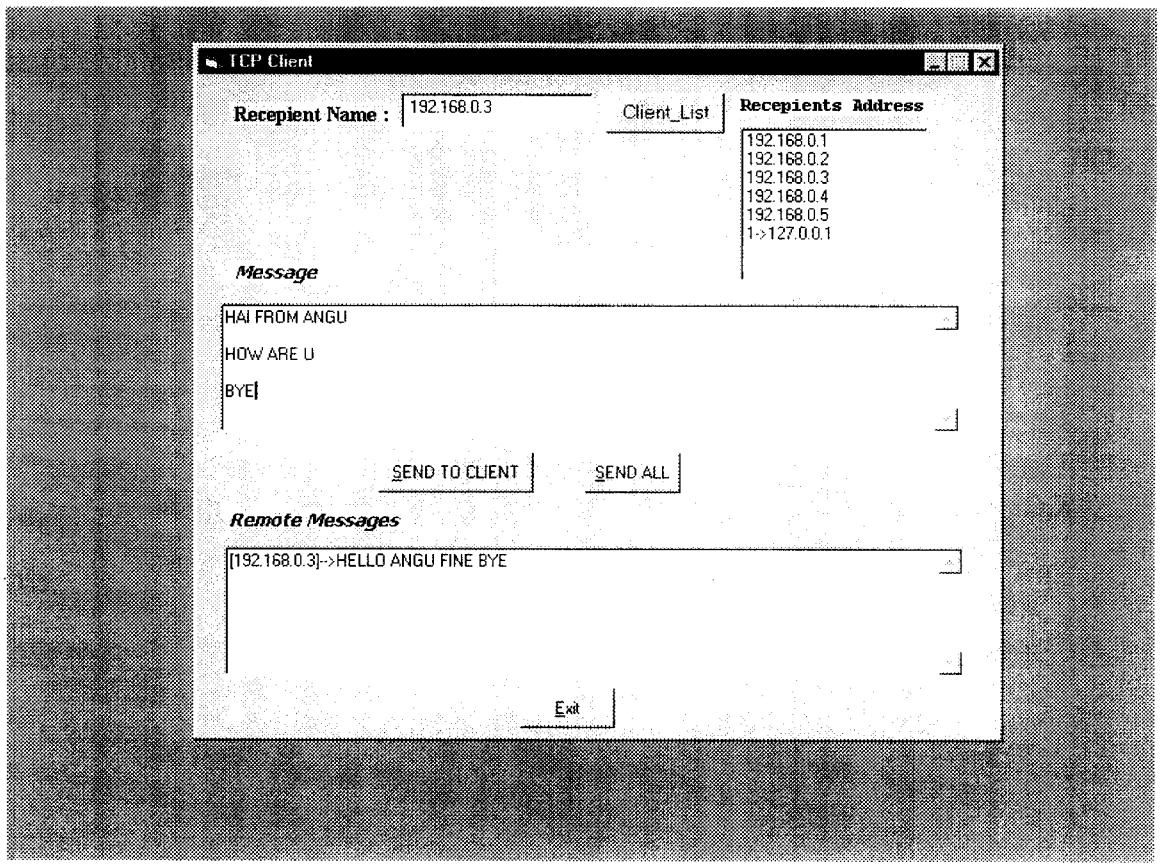
DBED Rs. 14600806 **BED Rs.** 6229163

Printed on Saturday, April 13, 2002


File Transfer
File Name :
Browse
Connect
Save
Quit
Received File Contents
Clear
Connection Information
Cannot Connect to RemoteHost
Wait for an incoming connection.....


File Name :
C:\My Documents\websites.txt
Browse
Send
Save
Quit
Received File Contents
file
File Successfully Sent
OK
Clear
Connection Information
*** Connected to IP Address:127.0.0.1 , Port #:5000


TCP Server
SERVER PORT
5000
Server Statistics
Server Name : ganesh
Server IP : 127.0.0.1
CLIENT IP ADDRESS
CLIENT MESSAGES
192.168.0.1
192.168.0.2
192.168.0.3
192.168.0.4
192.168.0.5
Close


TCP Client
Recepient Name :
192.168.0.3
Client_List
Recepients Address
192.168.0.1
192.168.0.2
192.168.0.3
192.168.0.4
192.168.0.5
1->127.0.0.1
Message
HAI FROM ANGU
HOW ARE U
BYE
SEND TO CLIENT
SEND ALL
Remote Messages
[192.168.0.3]-->HELLO ANGU FINE BYE
Exit

11. References

1) Visual Basic 6.0, Scott Warner, Tata McGraw-Hill Edition, 1999

2) Visual Basic 6.0 from the Ground up, Gary Cornell, Eastern Economy Edition, 1999

3) Oracle & Visual Basic Hand Book, SSI Group, SSI Material, 2000

4) **Network Programming for Microsoft Windows, Anthony Jones and Jim ohleund,** Tata McGraw-Hill Edition, 2000

5) Visual Basic 6.0 Complete Reference, Jerke, Tata McGraw-Hill Edition, 1999

6) Mastering Visual Basic 6.0, Evan Geros and Petroutsos, Tata McGraw-Hill Edition, 1999