MANAGEMENT OF CONCURRENCY PROBLEM IN SHARED DATABASE

:A CASE STUDY OF FIRST BANK NIGERIA PLC.

BY

UGBE, CATHERINE N. (Mrs.)
REG. NO : 98323036

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE AWARD OF POST GRADUATE DIPLOMA IN COMPUTER SCIENCE

DEPARTMENT OF MATHEMATICS, STATISTICS AND COMPUTER SCIENCE

COLLEGE OF SCIENCE AND AGRICULTURE

UNIVERSITY OF ABUJA.

SEPTEMBER, 2000

DEDICATION

This project work is dedicated to the Almighty God my father in heaven and to my trio Michael, Precious and Emmanuel.

ACKNOWLEDGEMENT

I wish to acknowledge the help and the intellectual support of my project supervisor Mr. F. Ogunfiditimi for making useful suggestion towards the successful completion of this work.

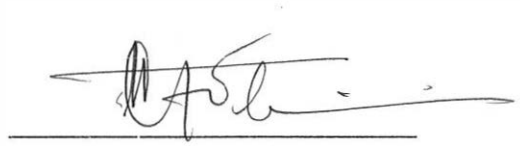
I also acknowledge the contribution of all the Lecturers to my Academic Success in the Department.

I am greatly indebted to all my friends for being there when their contributions were utmostly needed.

APPROVAL/CERTIFICATION

This is to certify that the work leading to the project was carried out by MRS. UGBE CATHERINE N.

The project has been certified as having met the requirements for the award of a Post-Graduate Diploma (PGD) in Computer Science.



29-9-2000

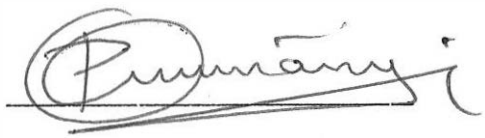
MR. F. OGUNFIDITIMI
PROJECT SUPERVISOR

DATE

3/10/2000

DR. E.O. OKU-UKPONG
HEAD OF DEPARTMENT MATHEMATICS,
STATISTICS AND COMPUTER SCIENCE

DATE



4-10-2000

EXTERNAL SUPERVISOR

DATE

ABSTRACT

A distributed database management system can be referred to as the physical storage and management of information in the database system that is distributed among separate computers within the network. An ideal distributed Database would be a "virtual filing system" one in which the physical location and retrieval of data is completely transparent to the user.

Distributed Database management system consist of many servers called database servers, which are software responsible for updating, deleting, adding, changing and protecting data which is usually stored on a hand disk to which many users on a network can have access.

Distributed databases have two key features which includes the ability to do a relational join with two tables from two database servers and the ability to write updates to two or more sites as part of one transaction using the Structured Query Language (SQL).

The Database server has the ability to regulate multiple simultaneous accesses via rules called concurrency controls thus preventing deadlocks. It also enforces security control rules to regulate access including data encryption.

The major advantages of this distributed database management system includes; the ability of geographically distributed organizations to cut down significantly their communication bills, reduced organizational impact that is, less bureaucracy, greater flexibility in operation, fast response to user enquiries and increased fault tolerance.

This research work will seek to develop a client server model, which is divided into front end and back end, the back end (or database server resides on a powerful machine and is responsible for number crunching data integrity and security cases. The front end resides on the users PC and is largely responsible for displaying data and serving the users queries to the back end.

Our major software developmental tool is Microsoft Visual FoxPro; this is a Relational Database Management system with Object Oriented Programming extensions on which vehicle the Structured Query Language (SQL) that will provide a major source of the database server attributes.

TABLE OF CONTENT

CHAPTER ONE

1.0 INTRODUCTION

1.1 Brief Introduction

Over the years, the role of the computer in banks has extended beyond that of enhanced routine data processing to include that of engendering the integration of decision making techniques with database and communication technologies to support storing, processing and retrieval of data over geographically dispersed locations.

Distributed Database management system has come to the fore in banking industry as the too; for managing applications that access data in multiple databases on various computer systems spread over geographically dispersed locations.

This tool has found good uses in banks because of the increasing need to manage data (customer account information) spread over its various branch network and the need to increase customer satisfaction in the competitive financial market of the twenty-first century, New Millennium.

The scope of this project work is limited only to the activities of the First Bank PLC, which is a first generation bank with a large network of branches, which are scattered nationwide.

The use of Information Technology in today's banking world has embraced the storing, processing and retrieval of information no matter where they are located and as such, it has become a popular quest in the bank's operational policy to truly maintain its market share and leadership.

Distributed Database management system has the attribute of an on-line, real time distributed processing that ensures the quick production of customer statements on demands or periodically, automatic computing of intense rates without stress, and customers ability to transact at any branch.

1.2 STATEMENT OF THE PROBLEM

The First Bank PLC and generally the banking world has over the years grappled with diverse kind of problems, which have greatly affected its ability

to perform creditably well amongst its competitors, especially among the New Generation banks. The later became fully computerized from inception. These problems include;

1. The world has become a global village where information can be got at the touch of a button. Thus, the inability of customers of the bank to get access to their account position accurately, fast from wherever is inevitable.

2. The inability of customers to lodge in money or other financial instruments or carry out transactions at any branch other than their local branch where the customer has an account.

3. The inability of customers to withdraw money at any branch.

4. The inability of banks management to get the financial position of the bank at any point in time.

5. Loss of main hour, which is usually, spent trying to manually reconcile inter-branch transactions.

1.3 OBJECTIVE OF THE PROJECT

The main objectives of this project include,

1. To increase productivity and profitability of the bank. This is expected that when the recommendations in this work are implemented the bank will have more customers and do more business.

2. To reduce the incidences of people moving huge sums of cash physically, consequently eliminates or reduce the loss of money to armed robbers.

3. To enable the banks management get an accurate financial position of each branch and ultimately the general bank's position at anytime.

4. To increase effectiveness and efficiency of the bank's operation.

1.4 JUSTIFICATION OF THE STUDY.

The use of this form of management information system through concurrency access can be justified thus,

1. Faster Response.

Customer wait time is greatly reduced, as staffs are able to attend to them better and faster.

2. Security.

Only authorized staffs are allowed access to certain files with the use of access code and appropriate status user identity.

3. Less Bureaucracy.

The management hierarchy associated with centralized system is greatly reduced because the front-end officers are given more powers.

4. Increased Motivation/Involvement of Users.

This system motivates more users and increase productivity.

CHAPTER TWO

2.0 LITERATUE REVIEW.

2.1 COMPUTERS IN ON-LINE AND REAL TIME PROCESSING.

Computers generally have gone beyond the days of batch processing, where data has to be collected on source documents and then accumulated and inputted at once.

On-line processing as defined by O'Leary and Williams (1985) refers to the case when a user's terminal is directly connected to the computer. However, that time processing is also known as transaction processing because transaction is processed so fast that the result can be used immediately. In transaction processing, data is inputted as it is created, this is the case in First Bank PLC.

On-line real time computers provide a very unique solution to the problem of distributed databases, especially for network data communication. This solution is aimed at providing data linkages amongst the Bank branches thereby making it possible to provide seamless services to its numerous customers nationwide.

2.2 DISTRIBUTED DATABASE SYSTEM.

Distributed Database systems are characterized by the ability to access data from geographically dispersed computers without the user necessarily needing to know where the data is located.

A potential advantage of distributed Database is that each database can be made the size appropriate for its amount of data, the complexity of user requirements and the number of users. As the system grows, added demands can be met more easily than with a centralized system. The update requires small changes to the existing databases or addition of new databases to the network.

Adigun (1992) opined that advances in the area of networking has given rise to database servers and its descendant distributed databases the advantage of this development is that geographically distributed organizations can cut down significantly their communication bills.

A typical distributed database system consist of a set of physical components, a set of sites, a set of transactions and system scheduler, an independent value is represented by a set of one or more (replicated physical component say y_1, y_2 ... which assume the same value except while update operation on them are in progress.

Basically, a component is a container of a data value. A logical database state is defined by a collection of data value such that exactly one data value is provided for each logical object in the system. The set of physical components in a distributed database system accommodates a subject of the set of physical components in the system, and each physical component belongs to exactly one site.

Page (1993) opined that Distributed Databases require the ability to move transparently with data regardless of where the data resides, since the system consist of many servers and many users, such a database has 2 key features;

(1) It is able to do a relational join with 2 tables, from 2 databases servers.
(2) It is able to write updates to two or more sites as part of one transaction.

This achieves the relational join feature by using a two-phase commit routine to handle multiple site updates.

2.2.1 DATABASE SERVER.

Adigun (1992) defines a Database server as a piece of software which is responsible for updating, deleting, adding, changing and protecting data which is stored on a hand disk to which many users on a network have access.
The structured form of stored data depends on the data model underlying the data base server, in order for the database server to regulate multiple simultaneous accesses; it exercises rules called concurrency controls. It also enforces security control rules to regulate access including passwords takes

advantage of the LAN's distributed architecture rather than co-existing with the LAN operating system as it is the case with traditional databases. The client-server model on which a database server is based, divides the database into a front-end and back-end.

The Back-end (or database-server) resides on a powerful machine and is responsible for number crunching, data integrity and security cases. The front end resides on the user's PC and it is largely responsible for displaying data and sending the users queries to the back end.

The front end provides the pieces needed for the user interface report writing, applications generators and other software associated with application generators.

Front end products include Microsoft Visual FoxPro, C, and Dbase IV etc., In summary there are 3 pieces in the arrangement, the database server or back-end manipulates, manages and protects data. The front-end software enters and displays data according to user's desires. SQL or some other query language is the mechanism of communication between the other two.

2.3 STUCTURED QUERY LANGUAGE (SQL)

The Structured Query Language is a well-defined set oriented data base foundational language that has been kept distinct from the procedural character of existing programming language.

The Structured Query Language is a well-known means of supporting relational databases and the relational approach; it can also be embedded in other host languages like Microsoft visual FoxPro. It is a language for interacting with relational databases.

SQL can be used in 2 primary ways, interactively through the command window as in the cases of Microsoft visual FoxPro and invoked from an application program. In both cases, the structure is the same.

SELECT : this is a list of data item of interest.

FROM: this is a list of tables where these data item exist.

WHERE: this is the condition(s) the data items must satisfy.

SQL is tool that allows relational join to be performed and also allow data at remote sites to be updated simultaneously.

2.3.1 VIEWS.

Views are an extremely powerful capability of the relational database approach. They create logical views of the data in addition to the base table. These views are then called the logical table, though they are not stored separately. Data can be stored in views for faster retrieval purposes. The views can then be used as tables in any SQL statement.

It is also possible to update a view by adding data to it and having that data update the base tables.

2.4 RELATIONAL DATABASE MANAGEMENT SYSTEM

Relational databases keep their data in a series of related flat files called tables. Use tables hold data in a row and column format with the rows corresponding to records and the columns corresponding to fields or data items.

The two most widely used operations in a relational database are the projections, which restricts data from he tables and the joints which combine different tables based upon a matching key field. The relational databases approach is the easiest to query but the slowest in processing transactions.

The table structure of a relational database system is simple and familiar. It is general enough to represent most type of data, it is independent of any internal computer mechanism and it is flexible because the user can restructure tables. Also the relational approach allows structured relationship between tables to be added later after the tables have been developed and data entered.

2.5 DATA COMMUNICATION.

Page (1993) also noted that Data communications and networking are increasingly needed by accounting information systems because transactions are recorded electronically, occur far from where they are analyzed and depend on data that are located far away.

Networks can provide a company with comparative advantages in increased speed of processing, better customer service, reduce cost and more complete analysis.

IBM first used data communications for transaction processing before the availability of microcomputers and developed a mainframe-oriented approach. This approach has major control advantages, as the processing takes place in the mainframe (host) and data communications are moved out to other devices.

2.5.1 NETWORK CONFIGURATIONS.

(a)Star: This consists of one or more small computers or peripheral devices connected to a large host computer or CPU. Star Networks are often used in a time-sharing system, in which several users are able to share a common central processor by receiving a time slice, which is a fixed amount of CPU time allotted to the users of CPU time allotted to the user's terminal.

Time-sharing can be established within a single organization or purchased from an outside service. Its advantages are:

(1) That it allows economical allocation of an expensive resource.

(2) That it allows geographically dispersed users access to a large, powerful system.

Its disadvantages are:

(1) That communications cost can be high for users in remote locations.

(2) That data may not be secure.

(b) Ring: This is a distributed data processing system, which is a decentralized computer system in which connected computers do their own processing in various locations, usually via telephone lines. This linkage allows the computers typically minicomputers, to share processing loads, programs, data and other resources.

Distributed data processing has several advantages.

(i) Processing requirements and demands are shared among several computers.

(ii) People at local sides are independent and have control over their own computer resources.

Its disadvantages include;

(i) There may be communication problems between CPU.

(ii) Data may not be secure.

(iii) More skilled programmers and computer operations than in a time-sharing, or centralized operation.

(iv) Tighter organizational control is required of all computing facilities.

(c) Bus: This is a connection of computers to a core wire, which can be either twisted pair or coaxial cable. When there is a problem on the line, the system cannot tell which computer is down.

Modem (short for Modulator/Demodulator): This is the device that converts digital signals to analog signals and vice versa. Modulation is the process of converting from digital to analog signal, and demodulation converts analog signal to digital ones. An acoustic computer is a cradle that allows the modem to accept a standard telephone handset.

2.5.2 NETWORK TYPOLOGY.

2.5.2.1 Local Area Network: A local area network is a direct high-speed connection of microcomputers that allows all of the communication needs just presented. LANS are typically formed from a group of stand-alone microcomputers whose users need to communicate. In addition to the microcomputers, a LAN typically requires the network interface cards, Network-aware software, and file-server and Network operating system.

2.5.2.2 Wide Area Network: When a network spreads over a large geographic area, it is often called Wide Area Network (WAN). Many WAN's consist of several types of computers connected worldwide. There are three-network

standard that can provide the backbone to sustain such a diverse network. These are, System Network Architecture, Transmission control protocol/internet protocol and open systems interconnect.

2.5.2.3 Internet: Simply put, the Internet is a large set of Computer Network that communicate with each other, over telephone lines. It enables companies, organizations, individuals and government to share valuable information across the world. The Internet includes World Wide Web, which enables you to see documents in richly formatted text and pictures.

2.5.3 CONCURRENT ACCESS.

This is another name for sharing files. It is safe to say that most applications will be run on LANS or some other forms of multi-user system.

Concurrency issue is generated when more than one person needs simultaneous access to a table. In database design consideration for multi-user operation influences a number of decision including the distribution of table e.g. the use of local files (files located on the hard disk of work stations rather than network services) and the careful management of the FoxPro data bytes.

Concurrency also has an impact on data security, data integrity and performance. The more people who use a system, the more opportunities exist for the data to be compromised.

Protecting files against unwanted access sudden hardware failure, and esoteric concurrency problems such as the deadly embrace (in which two users lock each other out of required table or record) need to be part of the design and utilization plan for the system.

In a multi-user system, you need to ask yourself a number of questions about each table;

(i) How many people will need to access it?
(ii) How often will it be accessed?
(iii) How important is it to keep the table available?

2.5.4 DEADLOCK

In a multi-user environment, this is the concurrency problem which can constitute roadblock in this environment, a deadlock occurs when one user has locked a record or a table and tries to lock another record that's locked by a second user who, in turn, is trying to lock the record that's locked by the first user. While such occurrences are rare, the longer that a record or table is locked, the greater the chance of deadlock

2.5.5 MULTI-USER SYSTEM

These are systems, which allow more than one user to access them, concurrently. Thus, in effect, more than one user can gain full access to its files without overwriting each other data.

CHAPTER THREE

3.0 SYSTEM ANALYSIS AND DESIGN

3.1 Introduction.

The main purpose of this chapter is to analyze the current system that is being implemented in the bank as at present and identifying the shortcomings with a view to preparing for the transition from the existing process to the new system and also to describe in details, the design of the proposed system.

System analysis and design is the process of examining and improving an information system and designing the improvements. It is basically a problem-solving approach and it consists of five steps:

1. Identifying a company's business information needs and determining the feasibility of meeting those needs.

2. Evaluating the effectiveness of company's present information system and specifying the new information requirements in details.

3. Formulating alternative information system and recommending the best of the systems.

4. Implementing the chosen information system.

5. Maintaining the new information system.

3.2 IMPORTANCE OF DISTRIBUTED DATABASE MANAGEMENT SYSTEM

Most commercial banks, First Bank in particular, have been going on over the years without computerization until recently, the following factors brought it to maintain its market leadership position, something urgent and adequate must be done about it. These includes:

(1) Organizational Growth : As the bank grew over the years, the need for timely information emerged and as such the bank needed to keep pace with its ever increasing list of customers demand of fulfilling their inter and intra banking operations.

(2) Better and Cheaper Technology: As both computer hardware and software are always decreasing in cost and improving in terms of efficiency, the bank is now better placed to justify its investment in the computerization project.

(3) Also the bank needs to meet up with the challenges posed by the so-called "New Generation Banks."

3.3 THE EXISTING SYSTEM

This system is also referred to as the manual system. It is an offshoot of the recording system applied in most conventional system, which involves a lot of paper work.

It involves operation such as:

(i) Opening of account
(ii) Depositing
(iii) Withdrawal
(iv) Interest payment.

3.4 IMPERATIVES FOR CHANGE

The manual system is bedeviled with a lot of problems which includes:

(1) Labour Intensive: The manual method is an energy sapping procedure, which also involved the management staff, thus withdrawing them away from formulating policies and implementing them to beat the competition.

(2) Time Wastage: In the business world, monetary terms and due to this fact, a lot of money that was made by the company has been lost by either looking for lost information or time spent in looking for a non existent information.

(3) The manual system over time develops what is known in administrative circles as Administrative Bottlenecks, which always lead to delay in processing an account because, somebody intentionally or otherwise did not enter the correct accounts information.

3.5 THE PROPOSED SYSTEM SPECIFICATION

Ogundana (1992) defined system specification as a detailed description of what the system is required to do. The proposed system helps the management to alleviate the problem of keeping track each customer accounts and reduce time wasted on trivial duties.

3.6 System Design

3.6.1 Aim

This is to present a solution to the inherent backlog of problems posed by the traditional method of operation in the bank. To achieve the above stated objective, a software package is developed which is relevant to the prevailing needs of the company.

3.6.2 Design Justification.

Ogundana (1992) noticed that the introduction of a computer-based solution to any business always comes with an incurred cost. But in about a year or two, the process of computerisation pays its cost over and over again in form of the time reduced in serving the needs of its customer, proper monitoring of each accounts etc.

3.6.3 Requirements and Cost Incurred

3.6.3.1 Software Involved

Microsoft Visual FoxPro (Version 6) software package, which is from the relational database language stock, is used to develop this Distributed Database management system for concurrent Access software. The major reason for using this application package is its ability to store relational databases and the ability to efficiently manage the storage requirements of the program and also because of its modulator programming capability, which allowed for this software to be written in manageable modules for easy comprehension and development.

This software can only be executed efficiently with the use of an operating system to produce results and the operating system that was used to develop this software is the Window 98

3.6.3.2 Minimal Hardware Requirements

- Client / Server Approach
- 1 server
  - ❖ Pentium III processor
  - ❖ 10.2 GB hard disk
  - ❖ Windows NT Server Operating System
  - ❖ 256MB of Memory (DIMMS)
  - ❖ SCS I Adapter Cards
  - ❖ SCS I drives e.g. floppy disk
  - ❖ Tape back up
  - ❖ 40 × CD ROM
  - ❖ 15'' SVGA Monitor
  - ❖ MODEM (Modulator / Demodulator)
  - ❖ Keyboard
  - ❖ 512 cache Memo
  - ❖ RAID (Redundant Array of Inexpensive Disk).
  - ❖ Network Interface Card
  - ❖ Workstations /Client

Workstations / Client

- 6.4 GB Hard disk
- 64 MB RAM
- 14'' Monitor
- Keyboard
- Network Interface card
- Windows 95/98 Client Operating System

Other Networking tools includes:

- ➢ RJ45 cabling
- ➢ Hub / Concentrator
- ➢ Repeater
- ➢ UPS
- ➢ Stabilizer
- ➢ Network Printer e.g. HP 4000N.

3.6.3.3 STAFF REQUIREMENTS

Since DDMS is a very important investment decision in a company, it will serve it well to provide it with a Database Administrator, Data Entry Operators, System Analyst and Network Engineer.

CHAPTER FOUR
4.0 SYSTEM IMPLEMENTATION

4.1 Introduction.

O'Leary and Williams (1982) defined system implementation as the set of activities that are performed to make the newly proposed and accepted system workable.

System implementation requires five basic activities to be very functional and these activities to be very functional and these activities are stated below:

(i) Software development
(ii) Hardware Acquisition
(iii) Personnel Training
(iv) Conversion

4.1.1 SOFTWARE DEVELOPMENT

This development was actually carried out using the Microsoft virtual FoxPro to develop the customized Software and it has been subjected to various tests to ascertain whether it has met all forms of requirements for the Computerization of the process of the organization.

4.1.2 HARDWARE AQUISITION

The hardware recommended for the Computerization of the bank has been bought and installed under the watchful eyes of a Systems Engineer and a Systems Analyst who supervised the software installation process.

4.1.3. PERSONNEL TRAINING

Majorly, the group of people who are to be trained are those people who are going to come in contact with the System on daily basis like the Data entry Operators, Database Administrator and the managers of the Company.

4.1.4 TESTING

After the hardware and the Software have been installed and the training had been in place, the various equipments (e.g. the Computer System) and the Software developed should be tested to know their capability and how useful they may be during operations.

Also, the operating procedure was tested. This involved a critical analysis of the procedure to see whether there will be any hostilities towards the use of

the new system. This actually developed when the administrative staffers felt they were being put out of work by the new system.

4.1.5 CONVERSION

This involves the process of changing over from the old information system implementation because the management were still skeptical about the whole idea of computerization due to the fact that they have not been fully weaned from the old manual system. However, there are four types of conversion and these include:

(a) Direct Conversion: This is the most risky type of conversion, which consists of completely discontinuing the old system and implement the new one. This approach should be avoided unless the organization is willing to live without any information should something go wrong with the new one.

(b) Parallel Conversion: The old and new information systems are operated side by side until the new one has shown that it is reliable, the old one is then discontinued. Since it is expensive operating the two systems simultaneously, the process should not be continued for long.

(C) Pilot Conversion: This involves using only a part of the organization to try out the new system i.e. a particular department could be used as the test site before the system is implemented throughout the organization. This approach is less risky than the direct method and less expensive than the parallel method, although it requires a long installation period to convert the whole organization to use the whole system.

(d) Phased Conversion: A new information system is implemented over a period of time, gradually replacing the old. Although it avoids the riskiness of some of the other approaches, it requires a long conversion time. This conversion process was chosen for this project to prevent any incident of risk.

4.1.6 PROGRAM FLOWCHART

Ogundana (1992) defined flowchart as a graphical or diagrammatical tool that shows the basic steps a particular program takes to accomplish a programming task. The program and the system flowcharts for this project is shown in Appendix A.

4.2 OVERVIEW OF THE SOFTWARE.

4.2.1 Managing Conflicts in Solving Concurrency problems

Whether you choose buffering, transactions, or views, you must manage conflicts during the update process.

1. Managing Buffering Conflicts

You can make data update operations more efficient by carefully choosing how and when to open, buffer, and lock data in a multi-user environment. You should limit the time a record or table is subject to access conflicts. Still, you must anticipate and manage the inevitable conflicts that result. A conflict occurs when one user tries to lock a record or table that's currently locked by another user. Two users cannot lock the same record or table at the same time.

Your application should contain a routine to manage these conflicts. If your application doesn't have a conflict routine, the system can lock up. A deadlock occurs when one user has locked a record or a table and tries to lock another record that's locked by a second user who, in turn, is trying to lock the record that's locked by the first user. While such occurrences are rare, the longer that a record or table is locked, the greater the chance of deadlock.

2. Trapping Errors

Designing a multi-user application or adding network support to a single-user system requires that you deal with collisions and trap for errors. Using Visual FoxPro record and table buffers simplifies some of this work.

If you attempt to lock a record or table already locked by another user, Visual FoxPro returns an error message. You can use SET REPROCESS to automatically deal with unsuccessful lock attempts. This command, in combination with an ON ERROR routine and the RETRY command, enables you to continue or cancel the lock attempts.

The following program called CONCURRENCY.PRG demonstrates automatic reprocessing of a failed operation, using SET REPROCESS.

Using SET REPROCESS and ON ERROR to Manage User Collisions

Code Comment

```
ON ERROR DO err_fix WITH ERROR(),MESSAGE() &&    This routine runs if an error occurs.

SET EXCLUSIVE OFF   && Open the files non-exclusively.

SET REPROCESS TO AUTOMATIC     && Reprocessing of unsuccessful locks is automatic.

USE c:\test\data\cust.dbf    && Open the table.

IF !FILE('cus_copy.dbf')
  COPY TO cus_copy       && Create the APPEND FROM table if needed.
ENDIF


DO app_blank  && The main routine starts here.
DO rep_next   && These commands are examples of codes that could be
                 executed in the course of your program.
DO rep_all
DO rep_curr
DO add_recs


ON ERROR     && This command traps the error
*!*      The main routine ends here.

PROCEDURE app_blank      && Routine to append a blank record.
  APPEND BLANK
RETURN
ENDPROC

PROCEDURE rep_next   && Routine to replace data in the current record.
  REPLACE NEXT 1 contact WITH ;
    PROPER(contact)
RETURN
ENDPROC
```

```
PROCEDURE rep_all      && Routine to replace data in all records.
  REPLACE ALL contact WITH ;
    PROPER(contact)
  GO TOP
RETURN
ENDPROC

PROCEDURE rep_curr     && Routine to replace data in the current record.
  REPLACE contact WITH PROPER(contact)
RETURN
ENDPROC

PROCEDURE add_recs     Routine to append records from another file.
  APPEND FROM cus_copy
RETURN
ENDPROC

*!*   Error Handling Using the ESC Key

*!*   Code Comment

PROCEDURE err_fix

  PARAMETERS errnum, msg

*!*   This program is called when an error is encountered and the user
escapes from the wait process.

DO CASE
*!*   Figure out what kind of error this is.
*!*   Is it "File is in use by another"?
  CASE errnum = 108
    line1 = "File cannot be locked."
    line2 = "Try again later..."

  CASE errnum = 109 .OR. errnum = 130
    line1 = "Record cannot be locked."
    line2 = "Try again later."
*!*   Or "Record is in use by another"?
```

```
  OTHERWISE
    line1 = msg + " "
    line2 = ;
      "See your system administrator."
ENDCASE

=MESSAGEBOX( line1 + line2, 48, "Error!" )
RETURN
*!*   Display the error message in a dialog box with an exclamation point
and an OK button.
```

3. Detecting and Resolving Conflicts

During data update operations, especially in shared environments, you might want to determine which fields have changed or what the original or the current values are in changed fields. Visual FoxPro buffering and the GETFLDSTATE( ), OLDVAL( ) and CURVAL( ) functions, enable you to determine which field has changed, find the changed data, and compare the current, original, and edited values so you can decide how to handle an error or conflict.

4. To detect a change in a field

After an update operation, use the GETFLDSTATE( ) function.

GETFLDSTATE( ) works on unbuffered data; however, this function is even more effective when you've enabled record buffering. For instance, use GETFLDSTATE( ) in the code of a Skip button on a form. When you move the record pointer, The Concurrency program checks the status of all fields in the record as below:

```
lModified = .F.
FOR nFieldNum = 1 TO FCOUNT( ) && Check all fields
  if GETFLDSTATE(nFieldNum) = 2  && Modified
    lModified = .T.
    EXIT && Insert update/Save routine here.
  ENDIF
ENDFOR
```

5. To detect and locate a changed record in buffered data

Use the GETNEXTMODIFIED( ) function.

GETNEXTMODIFIED( ), with zero as a parameter, finds the first modified record. If another user makes changes to the buffered table, any changes encountered by a TABLEUPDATE( ) command in your buffer will cause conflicts. You can evaluate the conflicting values and resolve them using the CURVAL( ), OLDVAL( ), and MESSAGEBOX( ) functions. CURVAL( ) returns the current value of the record on disk, while OLDVAL( ) returns the value of the record at the time it was buffered.

6. Solving Concurrency Problems Using The Locking Mechanism Approach

This is an effective means of keeping the level of Concurrency high and still prevent deadlock problems. The basic principle of locking is simple; if a user access a piece of data, the row will be locked and other users will not be able to access that row, only the user who has locked the row sees it.

Locks are released when the transaction ends, in other words, the life of a lock is never longer than that of the transaction in which the lock was created. Van der Lans(1993) noted that the locking mechanism works correctly if it meets the serialisability criterion. This means that the mechanism works correctly if the state of the database after(concurrently) processing a set of transaction is equal to the state of the database after processing the same set of transaction serially.

Locks administration is kept by this Database Server in the Internal Memory of the Computer, usually a large part of internal memory called the Buffer is reserved for this, thus indicating that locks are not stored in database

CHAPTER FIVE
5.0 CONCLUSION AND RECOMMENDATIONS

5.1 RESULTS

The aims of developing this software package were achieved. The results are listed below:

(1) A fully distributed Database Management System.

(2) On-line Analysis of various management reports.

(3) On-line banking capabilities.

(4) User-friendliness.

(5) Portability, that is, it can be used by varied commercial banks with similar requirements by making minimal changes.

(6) A full multi-user banking package.

(7) Efficient, that is, able to perform all the required tasks with high accuracy and minimum delay.

5.2 DIFFICULTIES ENCOUNTERED.

From the outset of this work, numerous difficulties were encountered and they could be broken down to the various stages they were encountered:

(1) This was the beginning of choosing the concept of this work.
It was noticed that after a lot of consideration and study of the company used in this project and their operations, it was decided that the service aspect of the business should be computerized because it is where the full effect of computerization will be felt immediately but disagreement ensued, but the concept later, won the day.

(2) Data Collection Stage:
People were first skeptical of the whole project because they thought it would go the way of all the other projects, which always had an enthusiastic beginning but fizzled out at the end of the day.

(3) Program Development Stage:

The project that cropped up here was what kind of application software to use in developing the package and of what stock or kind but at the end of the day, the Microsoft Visual FoxPro was chosen due to its ability to manage relational databases efficiently for its modular programming capability.

(4) Programming implementation Stage:

In this stage, the well-known fear of the unknown drifted in amongst the workers of the company because most of them were coming in contact with computers for the very first time and they were afraid of what it would do to them and their work, and as such they submitted data that were out of the touch with reality.

The implementation stage was fraught with error; but after a series of meetings by the top management with them they were able to see the benefits of computerization.

5.3 RECOMMENDATIONS:

The Software developed for this project work is capable of serving the basic needs of the bank as for now and into the nearest future.

However, it is recommended that the bank should acquire facilities that will allow customers with a PC and telephone lines to log into their server to access their bank account details and also to transact their business from the comfort of their homes.

Secondly, the bank should acquire data encryption software to prevent illegal access to the customer's account.

REFERENCES

Adigun M.O (1992) Distributed Database Management
COAN Proceedings of the XITH
National Conference, Enugu

Avison D.E (1991) Information Systems Development:
Methodologies, Techniques and Tools
Blackwell Scientific Publications Ltd, Oxford

Gehani N.O (1987) Concurrent Programming
Wellis Publishing House, England

Ogundana B.T (1992) Computers In the Society,
Onibonoje Publishers, Ibadan

O'Leary and Williams (1985) Computers and Information Processing,
Prentice-Hall Publishers, New York

Page J. (1993) Computerised Accounting Systems
Prentice-Hall Publishers, New York

Saxton J (1985) Distributed System: Concepts and Design
John Wiley Publishing Company,
Canada

Sommerville I (1990) Software Engineering
Addison-Wesley Publishing Company,
California

Van der Lans(1993) Introduction to Structured Query Language
Addison-Wesley, Amesterdam

Walton M (1998) Developer's Guide to Visual FoxPro
Programming
Microsoft Press, Atlanta Georgia

STRUCTURE OF DATA FILES

Structure for table: C:\PROJECT\DATA\CUST.DBF
Number of data records: 2
Date of last update: 09/27/00
Memo file block size: 64

Field	Field Name	Type	Width	Dec	Index
1	CUSTID	Character	10		Asc
2	PHOTO	General	4		
3	SIGN	General	4		
4	ACCTNUM	Character	16		
5	PRODID	Character	10		
6	PRODNAME	Character	50		
7	ACCTNAME	Character	50		
8	ODATE	Date	8		
9	HADDR	Character	50		
10	TEL	Character	9		
11	NATURE	Character	20		
12	REFREE1	Character	50		
13	HADDR1	Character	50		
14	REFREE2	Character	50		
15	HADDR2	Character	50		
16	MINWITH	Numeric	10	2	

* Total ** 442

Structure for table: C:\PROJECT\DATA\GENERAL.DBF
Number of data records: 4
Date of last update: 09/25/00

Field	Field Name	Type	Width	Index
1	CODE	Character	6	Asc
2	TYPE	Character	2	
3	DESC	Character	40	
4	OTHER	Character	20	

** Total ** 69

Structure for table: C:\PROJECT\DATA\TRANSACT.DBF
Number of data records: 0
Date of last update: 09/25/00

Field	Field Name	Type	Width	Dec	Index
1	TRANSACTIONID	Character	10		ASC
2	ACCTNUM	Character	10		
3	ACCTYPE	Character	10		
4	DC	Character	1		
5	TDATE	Date	8		
6	AMOUNT	Numeric	20	2	

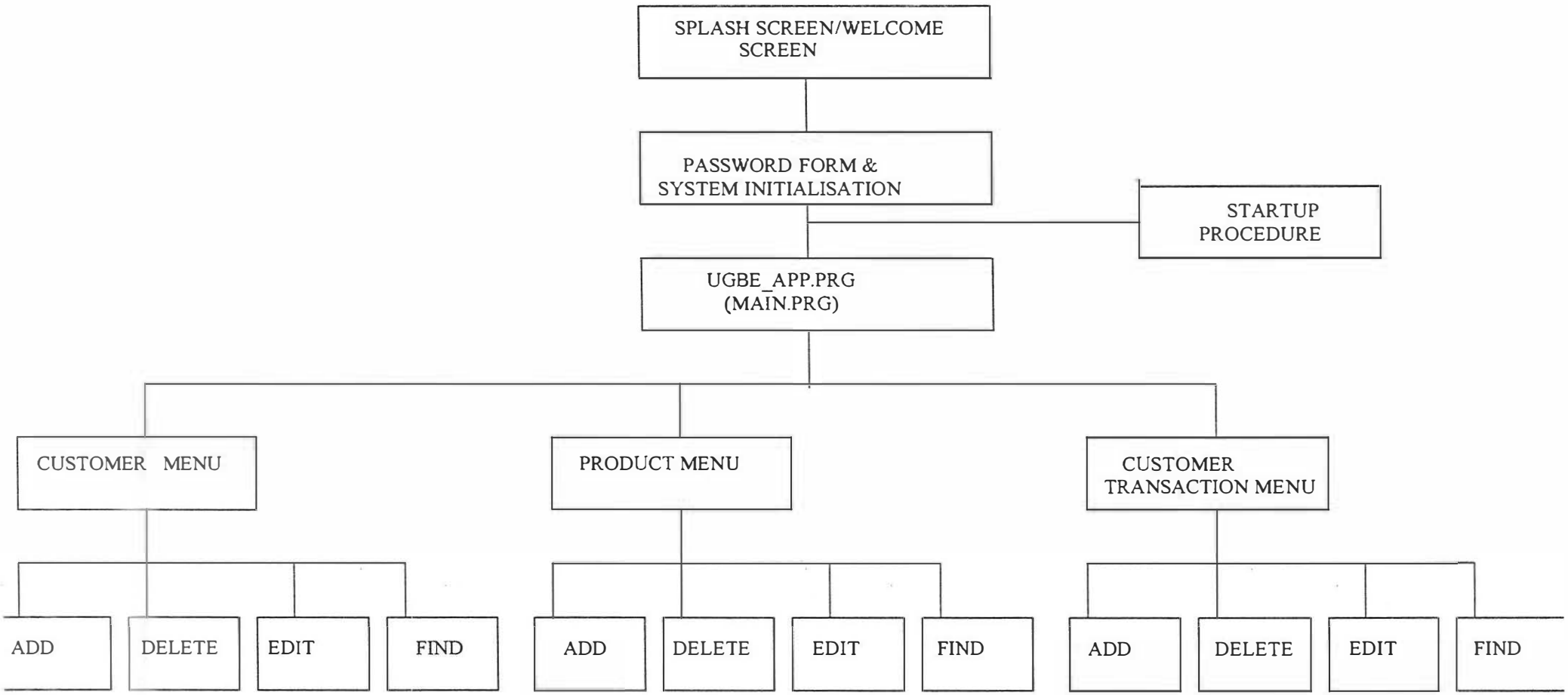
** Total ** 60

Structure for table: C:\PROJECT\DATA\SETUP.DBF
Number of data records: 5
Date of last update: 09/27/00

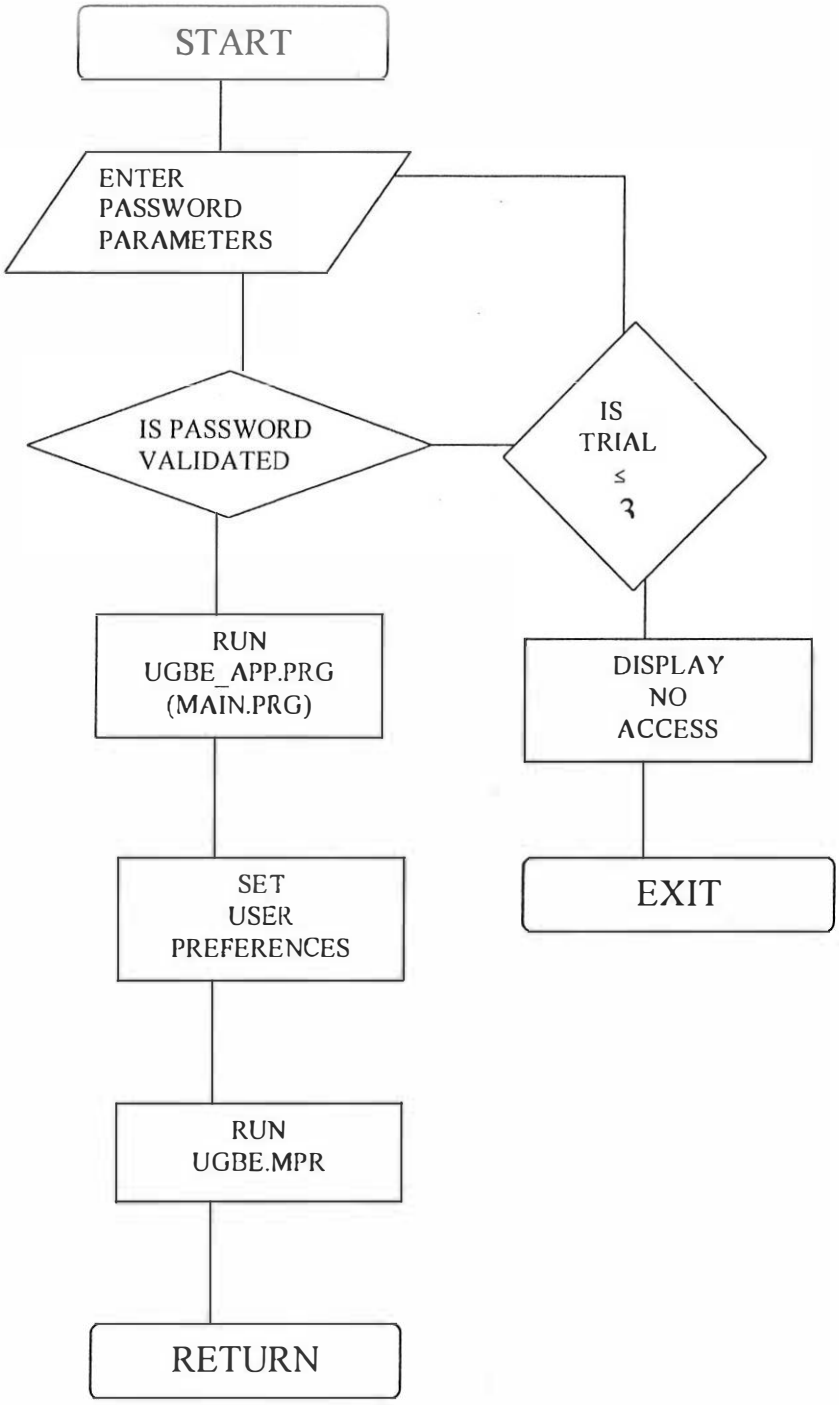
Field	Field Name	Type	Width	Dec
1	KEY_NAME	Character	20	
2	VALUE	Character	6	

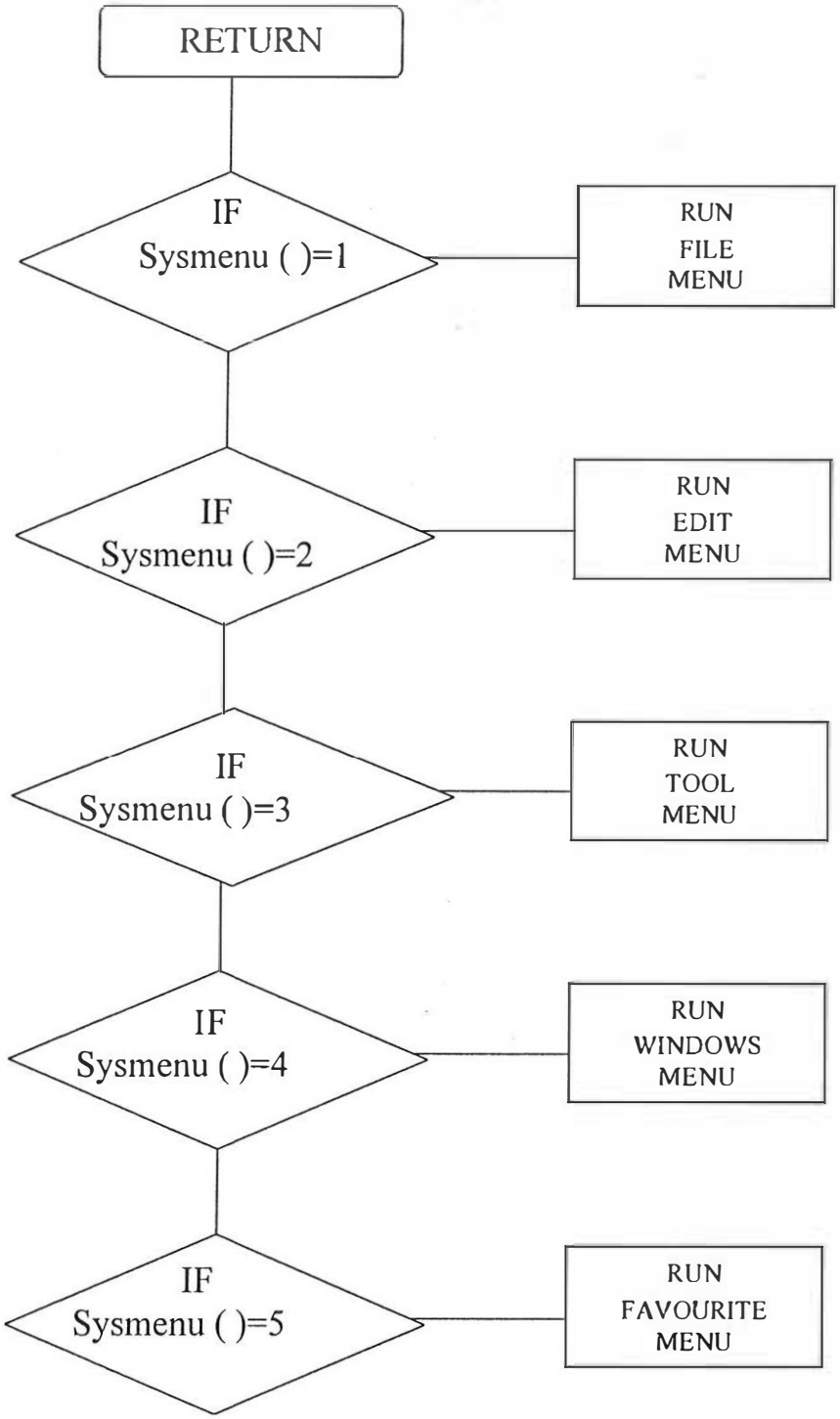
** Total ** 26

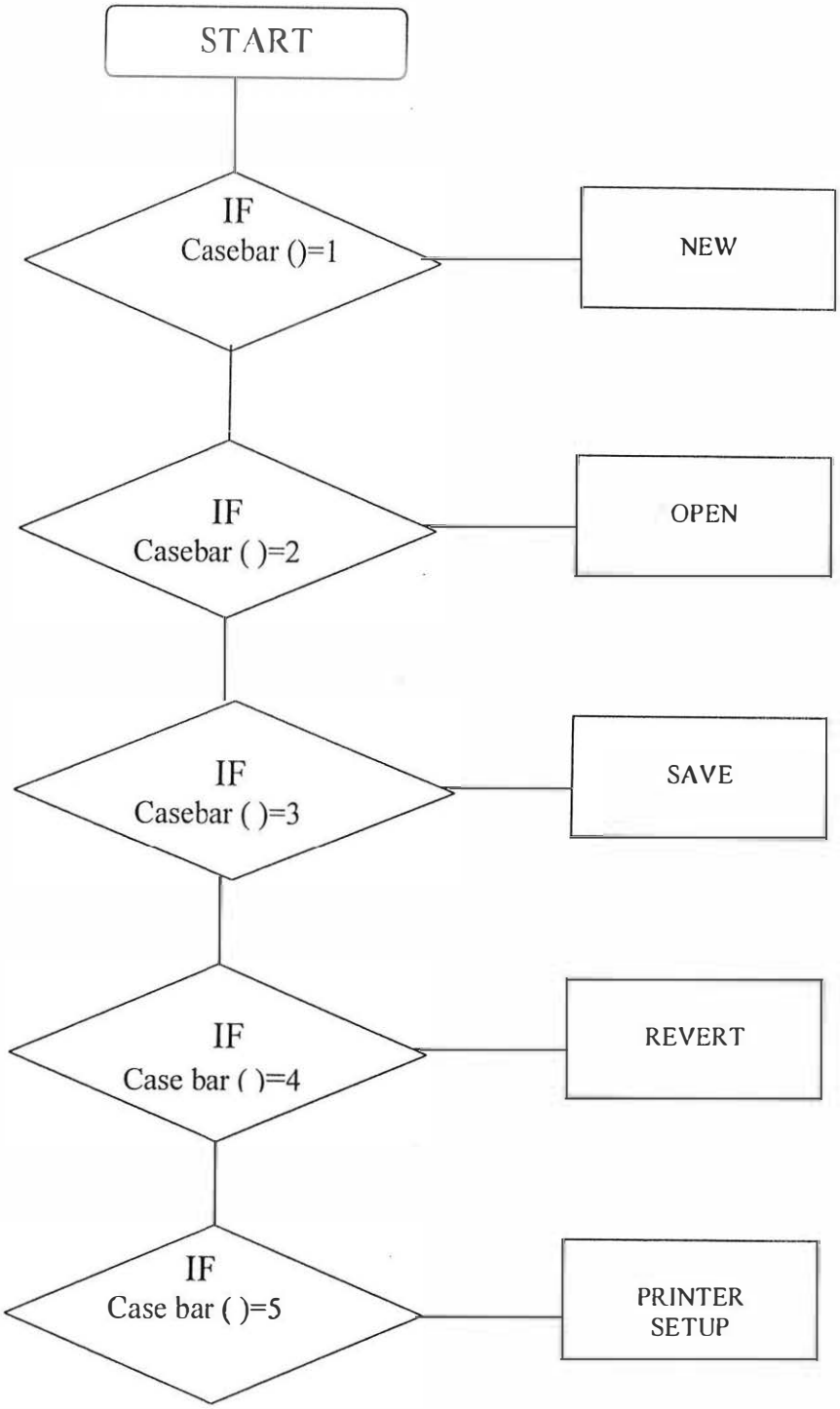
FLOWCHART

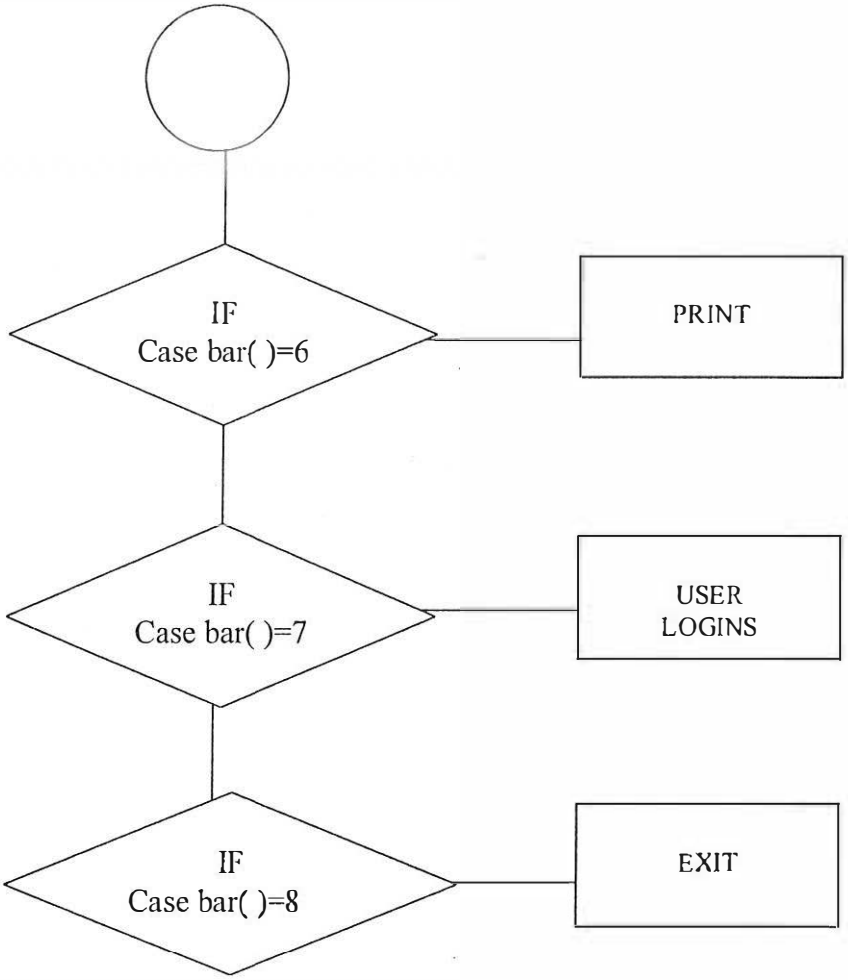

SPLASH SCREEN/WELCOME SCREEN
PASSWORD FORM & SYSTEM INITIALISATION
STARTUP PROCEDURE
UGBE_APP.PRG (MAIN.PRG)
CUSTOMER MENU
PRODUCT MENU
CUSTOMER TRANSACTION MENU
ADD
DELETE
EDIT
FIND
ADD
DELETE
EDIT
FIND
ADD
DELETE
EDIT
FIND

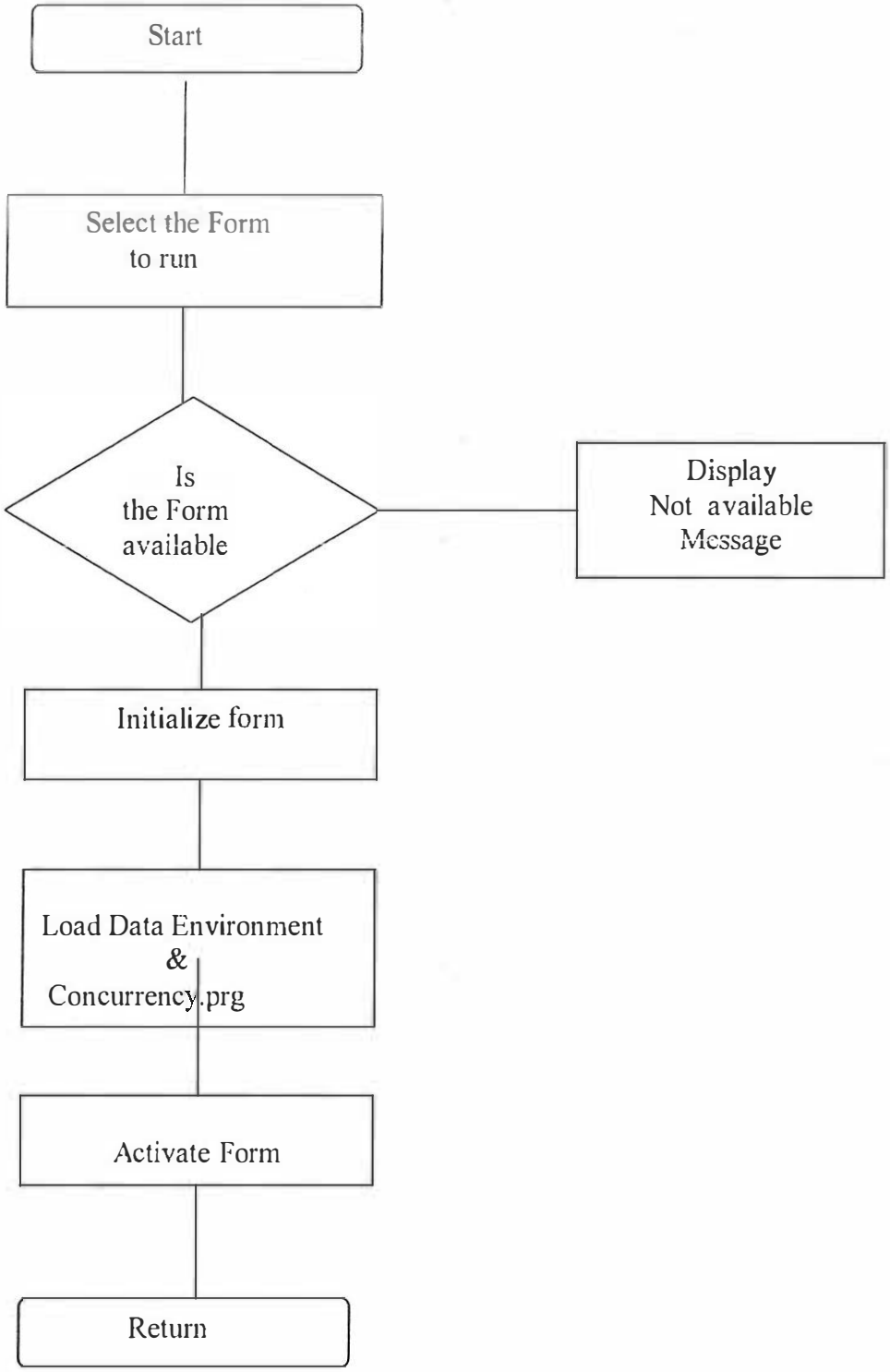
ACCESS & START-UP ROUTINE

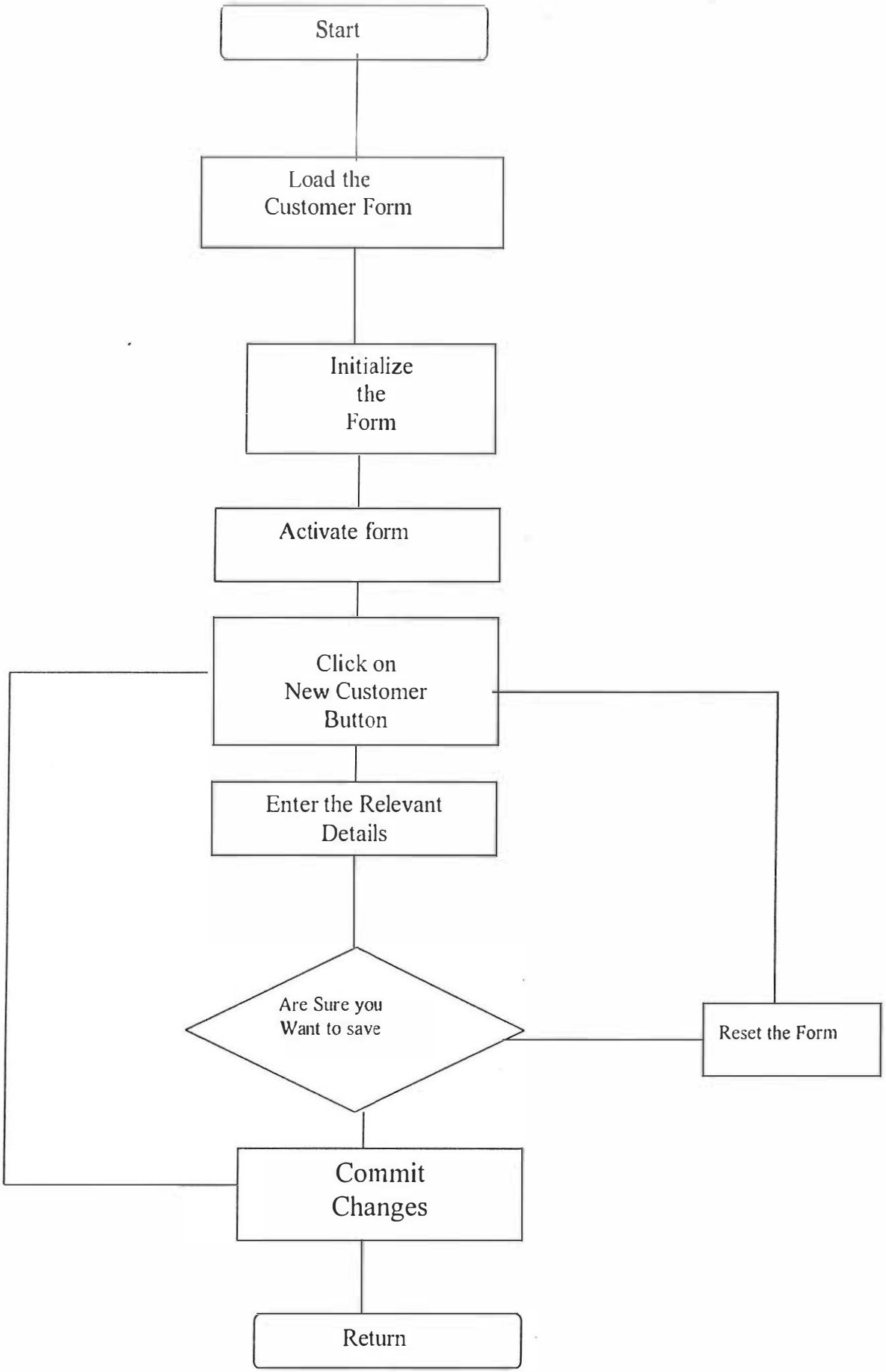



RETURN
IF
Sysmenu ( )=1
RUN
FILE
MENU
IF
Sysmenu ( )=2
RUN
EDIT
MENU
IF
Sysmenu ( )=3
RUN
TOOL
MENU
IF
Sysmenu ( )=4
RUN
WINDOWS
MENU
IF
Sysmenu ( )=5
RUN
FAVOURITE
MENU


START
IF
Casebar ()=1
NEW
IF
Casebar ( )=2
OPEN
IF
Casebar ( )=3
SAVE
IF
Case bar ( )=4
REVERT
IF
Case bar ( )=5
PRINTER
SETUP


IF
Case bar( )=6
PRINT
IF
Case bar( )=7
USER
LOGINS
IF
Case bar( )=8
EXIT


Start
Select the Form
to run
Is
the Form
available
Display
Not available
Message
Initialize form
Load Data Environment
&
Concurrency.prg
Activate Form
Return


Start
Load the Customer Form
Initialize the Form
Activate form
Click on New Customer Button
Enter the Relevant Details
Are Sure you Want to save
Reset the Form
Commit Changes
Return

PROGRAM LISTINGS

```
*!*    PROGRAM NAME: Transaction History Form
*!*    DESCRIPTION: This Program allows the user to browse through each
customer's past transaction
*!* OBJECT:COMBO1
*!* PROCEDURE:INTERACTIVE CHANGE


SET SAFE OFF
ThisForm.grdproducts.REFRESH
ThisForm.grdproducts.ROWSOURCETYPE=3
ThisForm.grdproducts.ROWSOURCE='SELECT
ALLTRIM(ACCTNUM),ALLTRIM(ACCTYPE),ALLTRIM(DC),TDATE,A
MOUNT FROM ugbe!transact WHERE
acctnum=UPPER(ALLTRIM(This.DISPLAYVALUE))INTO CURSOR
EMP_LEAVE'
ThisForm.grdproducts.COLUMNCOUNT=6
ff=UPPER(ALLTRIM(This.DISPLAYVALUE))
SELECT Transact.acctnum, Transact.dc, SUM(Transact.amount);
 FROM ugbe!transact;
 WHERE Transact.acctnum = FF;
  AND Transact.dc = 'D'INTO TABLE GG
USE GG
JJ=GG.SUM_AMOUNT
SELECT Transact.acctnum, Transact.dc, SUM(Transact.amount);
 FROM ugbe!transact;
 WHERE Transact.acctnum = FF;
  AND Transact.dc = 'C'INTO TABLE GG1
 USE GG1
JJ1=GG1.SUM_AMOUNT
KK=JJ-JJ1
thisform.text1.value=KK


*!*    FORM: Transaction History Form
*!*    DESCRIPTION: This Form allows the user to browse through each
                    customer's past transaction
*!* OBJECT:TEXT1
*!* PROCEDURE:CLICK
```

THISFORM.RELEASE

```
*!*     FORM: Customer Update Form
*!*     DESCRIPTION: This Form allows the user to update or add new
                     customer
*!* OBJECT: FORM1
*!* PROCEDURE: INIT

thisform.WIZFRAME1.Page1.Prodname1.VALUE='      '
thisform.WIZFRAME1.Page1.ACCTNAME1.VALUE='      '
thisform.WIZFRAME1.Page2.ODATE1.VALUE=date()
thisform.WIZFRAME1.Page2.HADDR1.VALUE='      '
thisform.WIZFRAME1.Page2.TEL1.VALUE='      '
thisform.WIZFRAME1.Page2.NATURE1.VALUE='      '
thisform.WIZFRAME1.Page2.REFREE11.VALUE='      '
thisform.WIZFRAME1.Page2.HADDR11.VALUE='      '
thisform.WIZFRAME1.Page2.REFREE21.VALUE='      '
thisform.WIZFRAME1.Page2.HADDR21.VALUE='      '
thisform.WIZFRAME1.Page2.MINWITH1.VALUE=0.00
thisForm.WIZFRAME1.Page1.CUSTID1.VALUE='      '
thisForm.WIZFRAME1.Page1.ACCTNUM1.VALUE='      '

*!*     FORM: Customer Update Form
*!*     DESCRIPTION: This Form allows the user to update or add new
                     customer
*!* OBJECT: FORM1.COMMAND1(NEW BUTTON)
*!* PROCEDURE: INIT


select cust
APPEND BLANK
thisForm.WIZFRAME1.REFRESH
thisform.WIZFRAME1.Page1.Prodname1.VALUE='      '
thisform.WIZFRAME1.Page1.ACCTNAME1.VALUE='      '
thisform.WIZFRAME1.Page2.ODATE1.VALUE=date()
thisform.WIZFRAME1.Page2.HADDR1.VALUE='      '
thisform.WIZFRAME1.Page2.TEL1.VALUE='      '
thisform.WIZFRAME1.Page2.NATURE1.VALUE='      '
thisform.WIZFRAME1.Page2.REFREE11.VALUE='      '
thisform.WIZFRAME1.Page2.HADDR11.VALUE='      '
thisform.WIZFRAME1.Page2.REFREE21.VALUE='      '
```

```
thisform.WIZFRAME1.Page2.HADDR21.VALUE='      '
thisform.WIZFRAME1.Page2.MINWITH1.VALUE=0.00

*!*    FORM: Customer Update Form
*!*    DESCRIPTION: This Form allows the user to update or add new
                    customer
*!* OBJECT: FORM1.COMMAND2(ADD BUTTON)
*!* PROCEDURE: CLICK

II1=ThisForm.WIZFRAME1.Page1.Prodname1.VALUE
II2=ThisForm.WIZFRAME1.Page1.ACCTNAME1.VALUE
II3=ThisForm.WIZFRAME1.Page2.ODATE1.VALUE
II4=ThisForm.WIZFRAME1.Page2.HADDR1.VALUE
II5=ThisForm.WIZFRAME1.Page2.TEL1.VALUE
II6=ThisForm.WIZFRAME1.Page2.NATURE1.VALUE
II7=ThisForm.WIZFRAME1.Page2.REFREE11.VALUE
II8=ThisForm.WIZFRAME1.Page2.HADDR11.VALUE
II9=ThisForm.WIZFRAME1.Page2.REFREE21.VALUE
II10=ThisForm.WIZFRAME1.Page2.HADDR21.VALUE
II11=ThisForm.WIZFRAME1.Page2.MINWITH1.VALUE

IF;
empty(II1).or.empty(II2).or.empty(II3).or.empty(II4).or.empty(II6).or.;
empty(II7).or.empty(II8).or.empty(II9).or.empty(II10)

cMessageTitle = 'Database Server Software'
cMessageText = "Please fill in all the Parameters Required"
nDialogType = 0 + 48
nAnswer = MESSAGEBOX(cMessageText, nDialogType, cMessageTitle)

ELSE

      cMessageTitle = 'Database Server Software'
      cMessageText = 'Do you want to save data postings'
      nDialogType = 4 + 32 + 256
      *  4 = Yes and No buttons
      *  32 = Question mark icon
      *256 = Second button is default
nAnswer = MESSAGEBOX(cMessageText, nDialogType, cMessageTitle)
```

```
DO CASE

        CASE nAnswer = 6

            SELECT cust
            REPL prodname with II1
            REPL acctname with II2
            REPL odate with II3
            REPL haddr with II4
            REPL tel with II5
            REPL nature with II6
            REPL refree1 with II7
            REPL HADDR1 with II8
            REPL refree2 with II9
            REPL haddr2 with II10
            REPL minwith with II11
thisform.WIZFRAME1.ACTIVEPAGE=1
thisform.WIZFRAME1.Page1.Prodname1.VALUE='      '
thisform.WIZFRAME1.Page1.ACCTNAME1.VALUE='      '
thisform.WIZFRAME1.Page2.ODATE1.VALUE=date()
thisform.WIZFRAME1.Page2.HADDR1.VALUE='      '
thisform.WIZFRAME1.Page2.TEL1.VALUE='      '
thisform.WIZFRAME1.Page2.NATURE1.VALUE='      '
thisform.WIZFRAME1.Page2.REFREE11.VALUE='      '
thisform.WIZFRAME1.Page2.HADDR11.VALUE='      '
thisform.WIZFRAME1.Page2.REFREE21.VALUE='      '
thisform.WIZFRAME1.Page2.HADDR21.VALUE='      '
thisform.WIZFRAME1.Page2.MINWITH1.VALUE=0.00

CASE nAnswer = 7
Delete
thisform.WIZFRAME1.ACTIVEPAGE=1

ENDCASE

ENDIF
```

```
*!*     FORM: Customer Update Form
*!*     DESCRIPTION: This Form allows the user to update or add new
                     customer
*!* OBJECT: FORM1.COMMAND3(CLOSE BUTTON)
*!* PROCEDURE: CLICK

THISFORM.RELEASE


*!*     FORM: Delete Customer Form
*!*     DESCRIPTION: This Form allows the user to delete  customer
*!* OBJECT: FORM2.TEXT1(SEARCH BOX)
*!* PROCEDURE: INTERACTIVE CHANGE


SET SAFETY OFF
SET DELETED ON


SELECT *;
 FROM ugbe!cust;
 WHERE cust.acctnum = ThisForm.Text1.VALUE;
   OR cust.acctname = ThisForm.combo1.VALUE;
   OR cust.prodname = This.VALUE;
   INTO CURSOR GGU
thisform.grdcusts.recordsource = "GGU"


*!*     FORM: Delete Customer Form
*!*     DESCRIPTION: This Form allows the user to close the form
*!* OBJECT: FORM2.COMMAND1
*!* PROCEDURE: CLICK

THISFORM.RELEASE
```

```
*!*     FORM: Edit Customer Form
*!*     DESCRIPTION: This Form allows the user to edit customer
                     information
*!* OBJECT: FORM3.TEXT1(SEARCH BOX)
*!* PROCEDURE: INTERACTIVE CHANGE


SET SAFETY OFF
SET DELETED ON


SELECT *;
 FROM ugbe!cust;
 WHERE cust.acctnum = ThisForm.Text1.VALUE;
   OR cust.acctname = ThisForm.combo1.VALUE;
   OR cust.prodname = This.VALUE;
   INTO CURSOR GGU
Thisform.grdcusts.recordsource = "GGU"


*!*     FORM: Edit Customer Form
*!*     DESCRIPTION: This Form allows the user to close the form
*!* OBJECT: FORM3.COMMAND1
*!* PROCEDURE: CLICK

THISFORM.RELEASE
```

```
*!*     FORM: Find Customer Form
*!*     DESCRIPTION: This Form allows the user to edit customer
                     information
*!* OBJECT: FORM4.TEXT1(SEARCH BOX)
*!* PROCEDURE: INTERACTIVE CHANGE


SET SAFETY OFF
SET DELETED ON


SELECT *;
 FROM ugbe!cust;
 WHERE cust.acctnum = ThisForm.Text1.VALUE;
  OR cust.acctname = ThisForm.combo1.VALUE;
  OR cust.prodname = This.VALUE;
  INTO CURSOR GGU
Thisform.grdcusts.recordsource = "GGU"

*!*     FORM: General  Form
*!*     DESCRIPTION: This Form allows the user to enter data on the
                     parameters of various kind because the software is
                     parameter-driven  information
*!* OBJECT: FORM4.TEXT1
*!* PROCEDURE: INTERACTIVE CHANGE

SELECT GENERAL

DO case
      CASE ThisForm.LABEL2.CAPTION='CODE'
            LOCATE for code=upper(alltrim(this.value))
            IF found()
                  thisform.Text2.value=type
                  thisform.Text3.value=desc
                  thisform.Text4.value=other
                  thisform.Label3.caption="TYPE"
                  thisform.Label4.caption="DESCRIPTION"
                  thisform.Label5.caption="OTHER INFORMATION"
```

```
            ENDIF
        CASE ThisForm.LABEL2.CAPTION='TYPE'
            GG=upper(alltrim(this.value))
            thisform.LIST1.ROWSOURCE='SELECT * FROM general
WHERE GENERAL.TYPE=GG INTO CURSOR UU'
            thisform.LIST1.ROWSOURCETYPE=3
            thisform.LIST1.COLUMNCOUNT=4
            thisform.LIST1.COLUMNLINES=.F.
        CASE ThisForm.LABEL2.CAPTION='DESCRIPTION'
            GG=upper(alltrim(this.value))
            thisform.LIST1.ROWSOURCE='SELECT * FROM general
WHERE GENERAL.DESC=GG INTO CURSOR UU'
            thisform.LIST1.ROWSOURCETYPE=3
            thisform.LIST1.COLUMNCOUNT=4
            thisform.LIST1.COLUMNLINES=.F.
        CASE ThisForm.LABEL2.CAPTION='OTHER INFORMATION'
            GG=upper(alltrim(this.value))
            thisform.LIST1.ROWSOURCE='SELECT * FROM general
WHERE GENERAL.OTHER=GG INTO CURSOR UU'
            thisform.LIST1.ROWSOURCETYPE=3
            thisform.LIST1.COLUMNCOUNT=4
            thisform.LIST1.COLUMNLINES=.F.
ENDCASE


*!*    FORM: General  Form
*!*    DESCRIPTION: This Form allows the user to enter data on the
                     parameters of various kind because the software is
                     parameter-driven  information
*!* OBJECT: FORM4.COMMAND1(DELETE)
*!* PROCEDURE: CLICK


  ThisForm.List1.visible=.f.
cMessageTitle = 'Database Server'
cMessageText = 'Are sure you want to delete this record ?'
nDialogType = 4 + 32 + 256
*  4 = Yes and No buttons
*  32 = Question mark icon
*  256 = Second button is default
```

```
nAnswer = MESSAGEBOX(cMessageText, nDialogType, cMessageTitle)
  ThisForm.List1.visible=.f.
DO CASE
  CASE nAnswer = 6
  ThisForm.List1.visible=.f.
  delete
  set dele on
  CASE nAnswer = 7
    ThisForm.release .
ENDCASE


*!*    FORM: General  Form
*!*    DESCRIPTION: This Form allows the user to enter data on the
                    parameters of various kind because the software is
                    parameter-driven  information
*!* OBJECT: FORM4.COMMAND2(CLOSE)
*!* PROCEDURE: CLICK

THISFORM.RELEASE



*!*    FORM: TRANSACTION  Form
*!*    DESCRIPTION: This Form allows the user to enter data on the
                    transaction between the bank and the customer
*!* OBJECT: FORM5.COMBO1
*!* PROCEDURE: CLICK

Set Deleted on
ThisForm.ListI.REFRESH
ThisForm.List1.ROWSOURCETYPE=3
ThisForm.List1.ROWSOURCE='SELECT acctnum,;
prodname,acctname,haddr FROM ugbe!cust WHERE;
PRODNAME=upper(ALLTRIM(This.DISPLAYVALUE))INTO CURSOR
EMP_LEAVE'
ThisForm.List1.COLUMNCOUNT=6
ThisForm.List1.REFRESH
ThisForm.List1.VISIBLE=.T.
```

```
*!*    FORM: TRANSACTION  Form
*!*    DESCRIPTION: This Form allows the user to enter data on the
                    transaction between the bank and the customer
*!* OBJECT: FORM5.COMMAND1
*!* PROCEDURE: CLICK

QQ1=ThisForm.TRANSACTIONID1.VALUE
QQ2=ThisForm.ACCTNUM1.VALUE
QQ3=ThisForm.RATE1.VALUE
QQ4=ThisForm.TDATE1.VALUE
QQ5=ThisForm.AMOUNT1.VALUE
QQ6=THISFORM.ACCTYPE1.VALUE
IF ;
empty(qq2).or. empty(qq3).or. empty(qq4).or. empty(qq5).or. empty(qq6)
cMessageTitle = 'Database Server'
cMessageText = 'Please fill in the required parameters'
nDialogType = 48
nAnswer = MESSAGEBOX(cMessageText, nDialogType, cMessageTitle)

ELSE

cMessageTitle = 'Database Server'
cMessageText = 'Do you want to save data postings?'
nDialogType = 291
nAnswer = MESSAGEBOX(cMessageText, nDialogType, cMessageTitle)
DO CASE
CASE NANSWER=6
select transact
appenD blank
replace transactionid with qq1
replace acctnum with qq2
replace DC with qq3
replace tdate with qq4
replace amount with qq5
replace acctype with qq6
CASE NANSWER=7
ENDCASE
```

```
cMessageTitle = 'Database Server'
cMessageText = 'Do you want to continue entering transactions for:'+' '+ qq2
nDialogType = 291
nAnswer = MESSAGEBOX(cMessageText, nDialogType, cMessageTitle)

DO CASE

CASE NANSWER=6

ThisForm.RATE1.VALUE='    '
ThisForm.TDATE1.VALUE='    '
ThisForm.AMOUNT1.VALUE='    '
ThisForm.Acctype1.VALUE='    '

CASE NANSWER=7
ThisForm.LBLTRANSACTIONID1.visible=.f.
ThisForm.TRANSACTIONID1.visible=.f.
ThisForm.LBLACCTNUM1.visible=.f.
ThisForm.ACCTNUM1.visible=.f.
ThisForm.LBLACCTYPE1.visible=.f.
ThisForm.ACCTYPE1.visible=.f.
ThisForm.LBLRATE1.visible=.f.
ThisForm.RATE1.visible=.f.
ThisForm.LBLTDATE1.visible=.f.
ThisForm.TDATE1.visible=.f.
ThisForm.LBLAMOUNT1.visible=.f.
ThisForm.AMOUNT1.visible=.f.
ThisForm.Command1.visible=.f.
ThisForm.Command2.visible=.T.
ThisForm.ComBO1.visible=.T.
ThisForm.LIST1.visible=.F.
ThisForm.COMBO1.VALUE='    '

ENDCASE

ENDIF
```

```
*!*     FORM: TRANSACTION  Form
*!*     DESCRIPTION: This Form allows the user to enter data on the
                     transaction between the bank and the customer
*!* OBJECT: FORM5.COMMAND2
*!* PROCEDURE: CLICK

THISFORM.RELEASE
```

CUSTOMER TRANSACTION REPORT

10/02/2000

Acctnum / Name	Date	Amount	Account-Type	Transaction Type
AC01Y2K123				
ADEDAPO ADEGOKE	09/22/200	156.00	SAVINGS	D
ADEDAPO ADEGOKE	09/27/200	56.00	SAVINGS	D
Subtotal for AC01Y2K123:		212.00		
Pct of subtotal:		46.28%		
Count for AC01Y2K123:				2
2				
AC01Y2K159				
UGBE CAHERINE	09/28/200	23.00	SAVINGS	D
UGBE CAHERINE	09/28/200	223.00	SAVINGS	D
Subtotal for AC01Y2K159:		246.00		
Pct of subtotal:		53.71%		
Count for AC01Y2K159:				2
2				
Total		458.00		
Pct of total		100.00%		
Total Count:				4
4				

GENERAL CODES REPORT

10/02/200

Type	Code	Desc
AC		
	AC001	SAVINGS ACCOUNT
	AC002	CURRENT ACCOUNT
	AC003	FIXED ACCOUNT
	AC006	LONG DEPOSIT

CUSTOMER PROFILE REPORT

10/02/200

Name	Account-Type	Account Number	Opening Date	House Address	Telephone
ADEDAPO ADEGOKE	SAVINGS ACCOUNT	AC01Y2K123	22/09/2000	Maitama	09-84848
UGBE CAHERINE	SAVINGS ACCOUNT	AC01Y2K159	25/09/2000	Wuse II	09-52306

CUSTOMER IDENTIFICATION REPORT

10/02/200

Customer Name	Account Number	Photograph	Signature
ADEDAPO ADEGOKE	AC01Y2K123		
UGBE CAHERINE	AC01Y2K159