

# **AUTOMATION OF CURRENCY PROCESSING**

(A Case Study of MICA Commercial Bank Plc. Maiduguri)

*By*

**CHARLES EJIKE ANYAEGBU**

**PGD/MCS/96/100**

A project Submitted to the Department of Mathematics/Computer Science, Federal University Of Technology, Minna, Niger State. In Partial Fulfilment of the Requirements for Award of Post-Graduate Diploma in Computer Science.

December, 1999

# **AUTOMATION OF CURRENCY PROCESSING**

(A Case Study of MICA Commercial Bank Plc. Maiduguri)

By

**CHARLES EJIKE ANYAEGBU**

**PGD/MCS/96/100**

Department of Mathematics/Computer Science  
School of Science and Science Education.  
Federal University of Technology, Minna.  
Niger State.

December, 1999

## CERTIFICATION

This project has been examined and found acceptable in partial fulfillment of the requirement for the requirement for the Post-Graduate in Computer Science of the Department of Mathematics/Computer Science of Federal University of Technology, Minna.

\_\_\_\_\_  
Prince Badmus R. O  
Project Supervisor

\_\_\_\_\_  
Date

\_\_\_\_\_  
Dr. S. A. Reju  
HOD Mathematics/Computer Science

\_\_\_\_\_  
Date

\_\_\_\_\_  
External Examiner

\_\_\_\_\_  
Date

## DEDICATION

This project work is dedicated to God the Almighty for His mercies throughout the period of this programme. To my family for their understanding throughout those weekends I had to leave them for the lectures.

## ACKNOWLEDGEMENT

First and foremost, I thank God for showering His blessing on me for achieving all what I have acquired and attained in my life.

My sincere gratitude to my project supervisor, Prince Badamasi Abdurashed for his untiring effort in making sure that this project becomes a dream come true.

My profound appreciation to the Head of Department, DR. S. A. REJU for his patience and understanding in the course of imparting knowledge on me. I thank you very much and may God Almighty reward you abundantly.

Also to my lecturers in the department for their patience, courage and understanding in course of imparting knowledge on me. I thank you all very much. May God in His infinite mercy bestow uncountable blessings on all, (Amen).

My deepest appreciation also goes to my entire family, my wife (Mrs Maureen Anyaegbu) and my sons (Okwuchukwu and Chibuikem) for their moral support, understanding, and courage especially at the initial stage of the programme when the going was a little bit tough. I love you all and thank you very much.

To friends, Mr. Wole Owosho and Engr. Peter Adaji for their untiring efforts in making sure that this project work becomes a reality.

Lastly, my regards to System Analyst – Olaiya Mohammed Mutiu of Ami-Tech Computers (Nig.) Ltd for his able criticism and support and encouragement in making sure that the overall total effort was not in vain. I thank you very much, may God in His infinite mercy reward you abundantly, (Amen).

## ABSTRACT

Presently, the processing/counting of currency (Bank notes) in commercial Banks in manual-based, that is, using counting machines, like the Glory and De-La Rue counting machines. The above process is often referred to as 'straight' counting because it only ascertains the number (100 B N in a packet) of Banknote in a packet of any denomination.

This machine cannot determine the genuineness of a banknote based on certain properties or features in a banknote. They do not sort bank into 'Fit,' Unfit' and 'suspect' note categories.

This project takes a close study of the Automation of currency processing in MICA Commercial Bank Plc, Maiduguri, with the aid of computer-based system.

At the end of the study, the Automated system would be able to sort banknotes into the categories stated above while counting them. It would also be able to determine the condition of banknote based on the properties and feature of each denomination.

If this study is completed successfully, MICA Commercial Bank would be able to handle the currency flowing bank from circulation in accordance with the Central Bank requirement in currency processing. It would also optimise their currency management system.

However, in order to achieve this, a customized software called ACP software will developed using the two most popular Database Management Packages (i.e *Dbase IV and Clipper 5.0*)

# TABLE OF CONTENT

Title Page -----	
Certificate -----	i
Dedication -----	ii
Acknowledgement -----	iii
Abstract -----	iv
Table of content -----	vi

## CHAPTER ONE CURRENCY PROCESSING

Introduction -----	1
The Links between Computer and the Counting Machine -----	1
Organogram of Mica Commercial Bank Plc Maiduguri -----	3
Function of Organogram -----	3
Project Motivation -----	5
Objective of the Study -----	5
Research Methodology -----	6
Basic System Concept -----	6
Scope of the Study -----	6

## CHAPTER TWO

### LITERATURE REVIEW

Functional Description of Currency processing -----	7
The Currency Management -----	7
The Existing Operation -----	8
The Problems Associated with the Existing Operation -----	8
Possible Solution to the Problems -----	8
Advantages of Currency Processing -----	9
Limitation of Manual – Based Currency Processing -----	10
Basic Features of Automated Currency Processing -----	11

## CHAPTER THREE SYSTEM ANALYSIS AND DESIGN

System Analysis, Design & Development -----	12
The Proposed Automated Currency Processing -----	12
Limitation of the Automated Currency Processing -----	12
The Structure of the New System (ACP) -----	13
The Main Menu System -----	13
Currency Manger Menu -----	13
Table Fill -----	13
Report Generation -----	13
Exit to Dos -----	13
The Output Format -----	17
The Input Format -----	20
File Design -----	21
Testing Project Feasibility -----	24
System conversion/Changeover Procedures -----	25
Cost Benefit Analysis -----	27

## **CHAPTER FOUR**

### **IMPLEMENTATION OF THE DESIGNED SOFTWARE**

Implementation of ACP -----	29
Choice of Programming Language -----	29
Hardware and Software Configuration Requirement -----	30
The ACP Installation -----	31
Test-Running/Troubleshooting of ACP -----	33
Starting the New System -----	33

## **CHAPTER FIVE**

### **SUMMARY AND CONCLUSION**

Summary -----	34
Conclusion -----	35
Recommendation -----	35
Reference -----	36



# CHAPTER ONE

## 1.0 INTRODUCTION TO CURRENCY PROCESSING

### 1.1 INTRODUCTION

The project is a case study of the automation of currency processing in MICA commercial Bank Plc Maiduguri. The automated system would be able to process and examine banknotes with the aid of sensors, which detect certain characteristics and properties of the banknotes. The system would also be in position to test banknotes on their condition and genuineness while counting them.

A sorting decision for each banknote is made by the machine's computer system. The dirty notes are stacked in the "unfit " area while the clean notes are stacked "Fit" compartment. The third groups of banknotes are those, which fail the genuineness, test and are thrown out for further inspection by the user/operator. The basic features of the system amongst others include an Errors free counting of notes. The performance characteristics would be discussed in details in Chapter three of this project work. The Automated Currency Processing (ACP) would no doubt enhance the existing note processing facilities in MICA Commercial Bank Plc with present sharp rise in the volume of currency-in-circulation, which has proved inadequate for their needs.

### 1.2 THE LINKS BETWEEN COMPUTER AND COUNTING MACHINE

Ever since the creation of computer, computation has been going on, then counting was made possible with the aid of finger tips, peddles and also by marks on the wall (as sign or symbols for counting).

As a result of advancement, the need then arises for a counting fingers tips and that of marks on the wall, as it became difficult to use method to sort and also count large sum of currencies.

This however, led to the advert and creation of life of the first counting machine called the ABACUS. The ABACUS is a counting machine which uses the peddles

to represent numbers and took the advantage of man's natural tendency to count in one s five (5fingers) and ten (5 fingers and 5 toes).

The ABACUS, thus it is designed to add, subtract and multiply numbers irrespective of whether the numbers are large or not.

This does not really solve the problems of counting and sorting machine for currency was invented.

- **THE GLORY COUNTING MACHINES.**

This machine is specifically designed to count and add currency together. It is Designed to use the principle of PASCAL (1645) and LEIBNITZ (1673) different Mechanical machine, which enhanced addition, subtraction and multiplication Techniques with an automatic carry transfer.

However, the glory counting machine does not possess the capabilities of sorting Fit, unfit and suspect note. The counting machines are mostly used in Commercial Banks in Nigeria.

- **THE DE-LA-RUE COUNTING MACHINE**

This is another counting machine for currency. It is the most common type of Counting machine found in most Nigeria commercial banks. It uses the same Principles as the glory counting machine. The machine cannot also sort fit, Unfit and suspect notes.

- **ISS 300PS MACHINE**

The ISS 300PS is a computer based currency-processing machine that is extremely more powerful than the later discussed and possesses those qualities that are lacked in the above counting machine.

However, some of its features include:

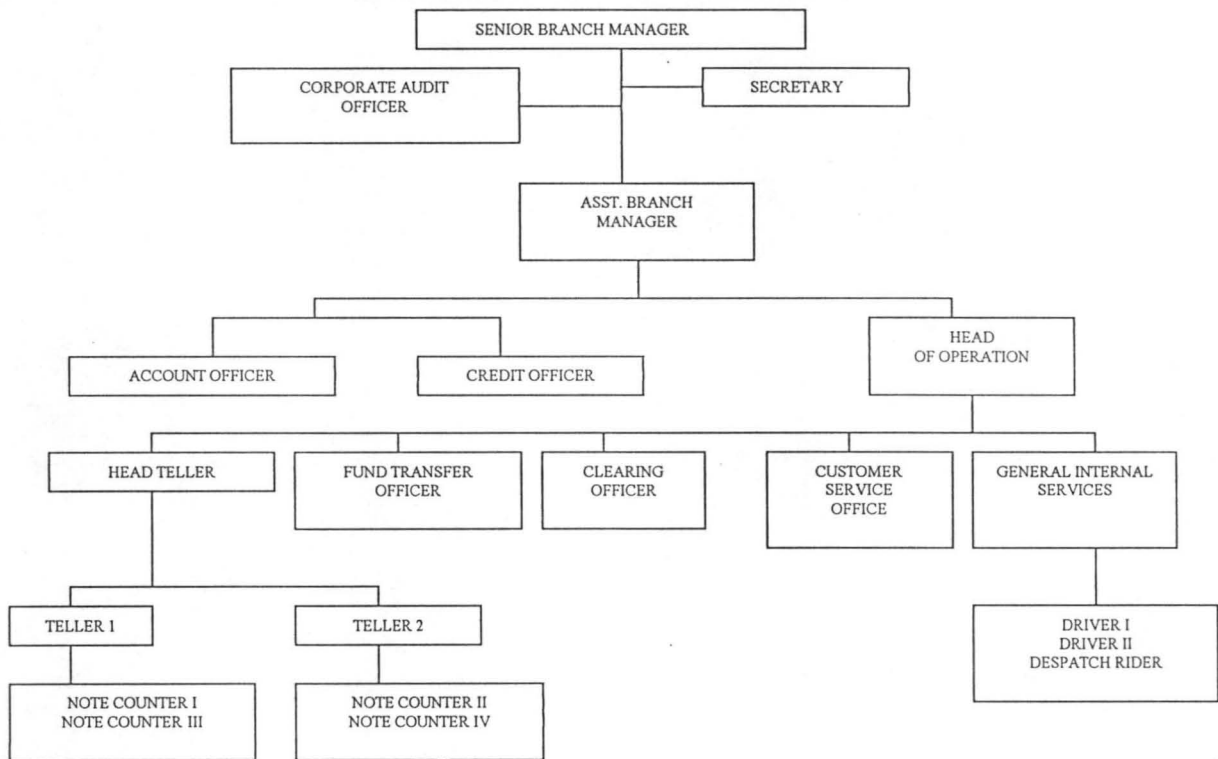
1. Adding, subtracting and multiplication of currencies.
2. Sorting while counting.

3. Sorting of fit, unfit and suspect bank notes.
4. Shredding of bank notes as desired.
5. Accounting and balancing after currency processing.

The ISS 300PS are mostly used by the Central Bank of Nigeria (the Apex Bank in Nigeria) because it is very expensive.

### 1.3 ORGANOGRAM OF MICA COMMERCIAL BANK PLC, MAIDUGURI

#### MICA COMMERCIAL BANK PLC MAIDUGURI BRANCH ORGANOGRAM



### 1.4 FUNCTIONS OF THE ORGANOGRAM

#### SENIOR BRANCH MANAGER:

This is the head of the branch. All other officers except the Audit Officer report to him through the Assistant Branch Manager. He attends quarterly Management meetings of the Central Management body at the Head Office. The administrative functions and the general performance of the Bank rest on him.

**BRANCH SECRETARY:**

This is the secretary of the branch responsible for all correspondence of the branch. Report directly to the Senior Branch Manager.

**CORPORATE AUDIT OFFICER:**

This unit carries out the internal audit exercise of the branch from time to time. They report directly to the senior manager.

**ASSISTANT BRANCH MANAGER:**

The assistant branch manager co-ordinates all the activities and the daily transactions of the bank and reports it to the Senior Branch Manager. Account officer, credit officer and head of operations reports to the assistance branch manager. He also acts for the Senior Branch manager when such cases arise.

**ACCOUNT OFFICER:**

The accounts officer is responsible for all internal functions such as staff salaries, contract payment and other staff functions outside the location.

**CREDIT OFFICER:**

The officer is responsible for all loan requests by staff and customers.

**HEAD OF OPERATORS:**

This officer is responsible for the entire customer related transactions e.g. Fund transfer, clearing customer services e.t.c.

**HEAD TELLER:**

The function of this officer is to confirm all cash lodgement already controlled by Tellers 1 and 2.

**FUND TRANSFER OFFICER:**

The officer is responsible for all transfer, cash or cheque within the location on inter-branch transfers by customer.

**CLEARING OFFICER:**

The function of this officer is to take all clearing instruments to the house for confirmation. The clearing exercise is carried out on daily basis at the Central Bank of Nigeria.

**CUSTOMER SERVICES OFFICER:**

This officer is responsible for new accounts e.g. Savings Accounts, Current Accounts e.t.c

**GENERAL INTER SERVICE OFFICER:**

This officer responsible for all in-house services to their officers.

**NOTE COUNTERS:**

All the deposits by customers are first of all counted and confirmed by the counters before lodgement. The note counters are responsible for the production of issuable currency to the customers. This unit is kind of mini-industry for the bank, for producing re-issue able notes. This case study is for the automation of its unit.

## **1.5 PROJECT MOTIVATION**

MICA Commercial Bank Plc, Maiduguri like any other bank comprises various departments including operations department. This department is responsible for the bank's cash transaction, which relates to the currency flowing back from circulation. This task of currency management could only be achieved through the Automated Currency Processing (ACP) since the manual –based processing had proved inadequate for their needs. Following a recent directive to all Commercial Banks by the Central Bank of Nigeria to “Process” their banknotes before depositing them with the Central Bank, it became necessary to undertake this study in order to automate their currency processing

## **1.6 OBJECTIVE OF THE STUDY**

MICA Commercial Bank Plc records enormous deposits of currency during its daily transaction from customers. There is also high level of withdrawals. In order to meet with these demands there is the need to process these notes to achieve a good stock of issuable

currency for withdrawal. The study will enhance the bank's customer services in terms of cash flows and currency management. Most importantly, the study has enormous potentials for establishing the cost of processing the currency in the Central Bank.

### **1.7 RESEARCH METHODOLOGY**

In order to achieve the above mentioned objectives; the study and analysis of the existing manual-based method had to be undertaken. It was then followed by the development of the Automated system to replace the existing facilities of the manual approach, which had proven inadequate for the bank. To accomplish this, the analysis of the system was carried out using interviews and record searching techniques for the proper understanding of the procedures of which the MICA COMMERCIAL BANK carries out currency management.

### **1.8 THE BASIC SYSTEM CONCEPT**

Below are the Basic system concepts of the proposed Automated Currency Processing

- ✓ The system needs one operator/user at a time. This will clear delimitation of the responsibility for the banknote processing operation.
- ✓ Correlation between depositor's Bank and processed note pack is maintained at any time to enable discrepancies to be traced back.
- ✓ Denomination changes by simple keyboard entry.
- ✓ Highly flexible hardware and software architecture to meet all requirements.
- ✓ Capacity of processing bad quality notes
- ✓ Modular system for ease of maintenance
- ✓ High technical standard to ensure optimum availability
- ✓ Designed for operator/user convenience.

### **1.9 SCOPE OF THE STUDY**

The Project work is intended to cover the currency processing operation of the MICA Commercial Bank Plc, Maiduguri. Specifically, the area of interest is the transformation of the manual based record keeping of currency processing into the Automated currency processing (ACP) which is going to be a Computer Dos-Based system.

## CHAPTER TWO

### 2.0 LITERATURE REVIEW

#### 2.1 FUNCTIONAL DESCRIPTION OF CURRENCY PROCESSING

Banknotes are processed and examined by the Automated system with aid of sensors, which detect certain characteristics and properties of the notes. The system is in a position of testing notes on their condition and genuineness while counting them. The sensors transfer the result of their findings to the machine computer system that compiles and evaluates them. A sorting decision for each individual banknote is made by the machine computer system. A banknote is then stacked accordingly in its appropriate compartment. During this process, a reject sensor rejects some notes with missing or folded parts. These notes are to be re-examined by the operator/user and then feed back into the system. The fake notes or those that do not meet the requirement are sent to a separate compartment (manual inspection compartment).

#### 2.2 CURRENCY MANAGEMENT

The role of currency processing in commercial banks is to ensure optimal currency management system. The adoption of the study on the Automatic currency processing system in commercial banks is therefore designed to replace the existing facilities, which are largely manual-based. With the manual operation, a bank could hardly produce required currency outflow in a day. The quality and originality of the banknotes produced by this manual-based system are not assured. The manual system cannot detect a suspect note and cannot sort currency into "Fit" and "Unfit" categories. This study has an enormous potential for stabilizing the cost of producing and distributing the currency. It will enable commercial banks to have a good grip on currency management system and to optimize it.

In order to adopt a good currency management system, the study takes the following features into consideration.

- ✓ Speed of sorting currency
- ✓ Ability to sort while counting
- ✓ Ability to detect suspect note
- ✓ Ability to separate "dirty" note and "clean" notes

### 2.3 THE EXISTING OPERATION

The existing method of operation that is processing of currencies at MICA Commercial bank Plc, Maiduguri is completely manual-base. The bank uses a mechanical counting machine called the De-La-Rue counting machine of the bank notes transaction for each day. This method of operation is also known as the “**STRAIGHT COUNTING**” which only involves counting of 100BN (banknote) for a packet of any denomination.

### 2.4 THE PROBLEM ASSOCIATED WITH THE EXISTING OPERATION

Even though the De-La-Rue counting machine proves very efficient in counting of Bank notes, certain problems are associated with the manual-based machine operation. Few among the problems are as follow:

- (1) De-La-Rue counting machine cannot sort currencies to different denomination.
- (2) De-La-Rue counting machine cannot sort bank notes based on clean and unclean (that is fit, Unfit and Suspect Bank notes).
- (3) The De-La-Rue is an Electro-Mechanical machine that does not have micro-processor, hence it cannot store information as regards any note
- (4) Sorting fit, Unfit and fake bank notes are done manually, thereby slowing down daily transaction.
- (5) Often at times, the De-La-Rue counting machine does make some mistakes in counting especially when there is an unfit bank note.

### 2.5 POSSIBLE SOLUTIONS TO THE PROBLEMS

Going by the above aforementioned problems, the following possible solutions to the problems are recommended:

1. A computer based counting machine like the ISS-300PS should be acquired, or a machine that can sort out currencies based on denomination, Fit Unfit and Suspect (fake) bank notes.
2. A computer-based software called “Automated Currency Processing (ACP)” will be developed to allow storing of currency processing information.



3. The ACP software will also allow data entry to be made possible on the daily transaction for fit, Unfit and Suspect (fake) bank notes, based on the denomination.

Moreover, following a recent directive to all commercial banks in the federation by the Apex bank to “*Process*” their bank notes before depositing them with the Central Bank of Nigeria. The Automated Currency Processing (ACP) software is the best computerised method to be used in solving these problems.

## 2.6 ADVANTAGES OF CURRENCY PROCESSING

With the enormous demand of issuable currency in Commercial banks and in order to meet the demand and to be able to handle the currency flowing back from circulation currency processing remains a vital operation in Commercial banks.

It is more regarded as mini-industry for producing issuable notes as an alternative to mint notes (new banknotes).

The notes produced by this process are genuine and are sorted according to “Fit”, “Unfit” and “Suspect” note categories. The process is also self-accounting or balancing, on basis of 100 processed notes. It can also balance on the basis of any desired deposit value comprising different denomination and depositors.

Furthermore, processing results are available on hard-copy printouts formatted according to individual user requirements.

In summary, the following advantages stand to be gained from the ACP software, they include: -

- It is a Mini-bank Industry for producing issue-able bank notes as an alterative to mint notes that comes from the Apex bank (CBN).
- The processing of currency is self-accounting or balancing, on the basis of 100 bank notes processed.

- The Automated currency processing software (ACP) will allow adequate reports to be generated based on the following:
  1. Report on types of banknotes denomination.
  2. Report on fit note received from the customers on a daily transaction.
  3. Report on unfit bank noted received from the customers on daily transaction.
  4. Lastly, it increases efficiency in transaction with minimum human energy and available resources.

## 2.7 LIMITATIONS OF MANUAL-BASED CURRENCY PROCESSING

Manual-based currency processing has a lot of limitations. In this present day requirements of currency circulation, the standards of currency outflow from commercial banks are enormous. The limitations of the manual-based currency processing in this regard are stated below:

- ❖ Inability to detect suspect notes
- ❖ Inability to differentiae between clean and dirty notes
- ❖ Low speed of processing
- ❖ No self accounting facility
- ❖ No sorting of notes in any form
- ❖ Cannot determine the genuineness of notes based on certain properties and features of the notes
- ❖ Cannot meet with the currency flowing back from circulation

This study will enhance the existing Note processing facilities in commercial banks, which in the face of the sharp rise in the volume of currency in circulation has proven to be inadequate for their needs.

## **2.8 BASIC FEATURES OF AUTOMATED CURRENCY PROCESSING**

The Automated currency Processing Software (ACP) is going to be a computer-based system and program with the following features.

1. The ACP has the ability to sort currency based on the type of domination (i.e. N10, N20 and N50).
2. Ability to sort currency based on whether the bank notes is fit, Unfit and suspect/fake.
3. Ability to sort currency while counting is going on.
4. Ability to detect fake/suspect bank notes.
5. Generation of necessary report by the ACP for management decision-making.
6. Ability to store large volume of currency information.
7. ACP provides flow of data during data entry stage.
8. It is year 2000 (Y2K) compliant.
9. It is only accessible by the authorized users.

# CHAPTER THREE

## 3.0 SYSTEM ANALYSIS

### 3.1 INTRODUCTION

System Analysis can be defined as the process of examining an existing system in an organization. The person examining the existing system is called the system Analyst.

Specifically, the duty of System Analyst is usually to carry out the following task.

- Adequate feasibility study
- Analysis of an existing system
- Design of a computer based system (program)
- Implementation and review of developed system.

On the other hand, system design and development covers all design and development activities of an information system within an organization like the MICA Commercial Bank Plc, Maiduguri. The activities here involved the analyst and the programmer and in most cases, the analyst becomes the leader of the team because of his wide knowledge in system design and program (software) development. Their major job includes design of new system (ACP), development of new system and maintenance of an existing system.

MICA Commercial Bank Plc, Maiduguri does not possess an existing system, the current mode of processing currency is a manual based method which therefore call for a computer based program called Automated Currency Processing (ACP).

### 3.2 THE PROPOSED AUTOMATED CURRENCY PROCESSING

The proposed ACP is going to be a computer-based program, flexible easy to use and understand. The analysis, design and program development of the ACP was based on top-down modular programming procedure. This shows that the existing operation was broken down from complex to simple procedure in order to treat and design the new system (ACP) in modular form.

### 3.3 LIMITATIONS OF THE AUTOMATED CURRENCY PROCESSING

Every design and developed program has a shortcoming especially when the developed program is an integrated program, and perhaps most standard application software packages are integrated program.

Integrated packages are packages that incorporate several other features of stand alone programs like spreadsheet, Database Management System, Word processing, Graphics and communication software into one program as a stand alone software thereby causing unbalance maximum utilization. Its because of this reason that developed software are released in *versions* and *series* in order to update current and relevant information in the system from time to time.

Because of the above reasons, the following shortcoming and limitation are envisaged:

- ❖ The ACP is designed and developed to be used only by the case study (MICA Commercial Bank Plc, Maiduguri)
- ❖ It is designed to run on a stand-alone computer system.
- ❖ It is limited only to authorize users. Authorized users are those users who have ultimate full access to the ACP software.

### 3.4 THE STRUCTURE OF THE NEW SYSTEM (ACP)

The structure of the new system (ACP) is designed to be user - friendly with an active user interface. An active user interface system is a system that is easy to understand and menu driven. The structure of ACP is divided into five subs - sections as follows:

#### 3.4.1 THE MAIN MENU SYSTEM

The main menu system served as an entrance to all modules of the system. It is designed to use numeric digit and arrow keys to highlight end - user / operators desired option with a press on the ENTER key to execute the procedural program concerned. Specifically, the main menu comprises the following options:

#### CURRENCY MANAGER

- TABLE FILE
- REPORT GENERATION
- EXIT

Below is a diagrammatical representation of the main menu of the ACP.s

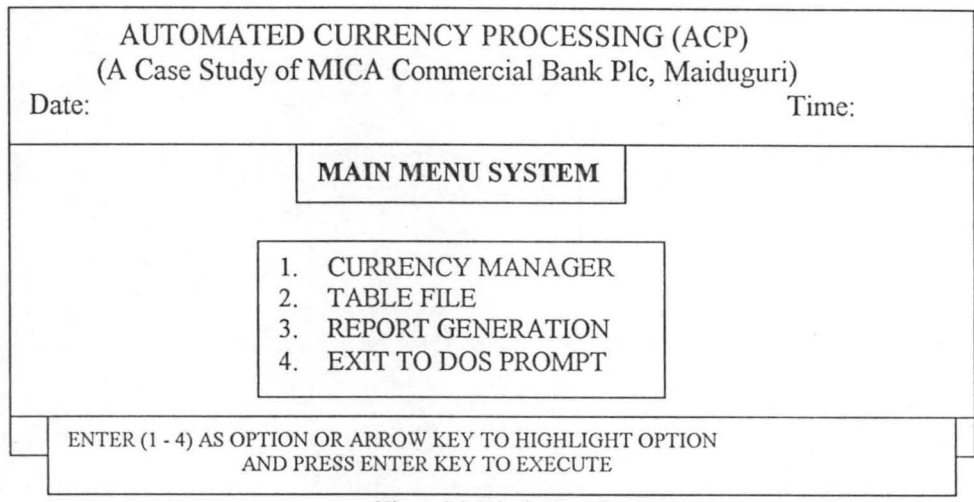


Fig 1.0 Main Menu System

### 3.4.2 CURRENCY MANAGERS

The Currency Manager is the heart of the automated currency processing. This is where all currency processing takes place.

Here currency are processed based on the type of denominations, clean bank notes (FIT) are sorted from the Unclean (UNFIT) and consequently the fake / suspect bank notes are also sorted out.

However, in order to achieve a good result from the sorting process, a powerful program routine (procedure) was designed and developed to take care of this sorting process. The routine program is called *Currency Processing*.

The general format of the CURRENCY MANAGER, is as shown below in Fig 2.0

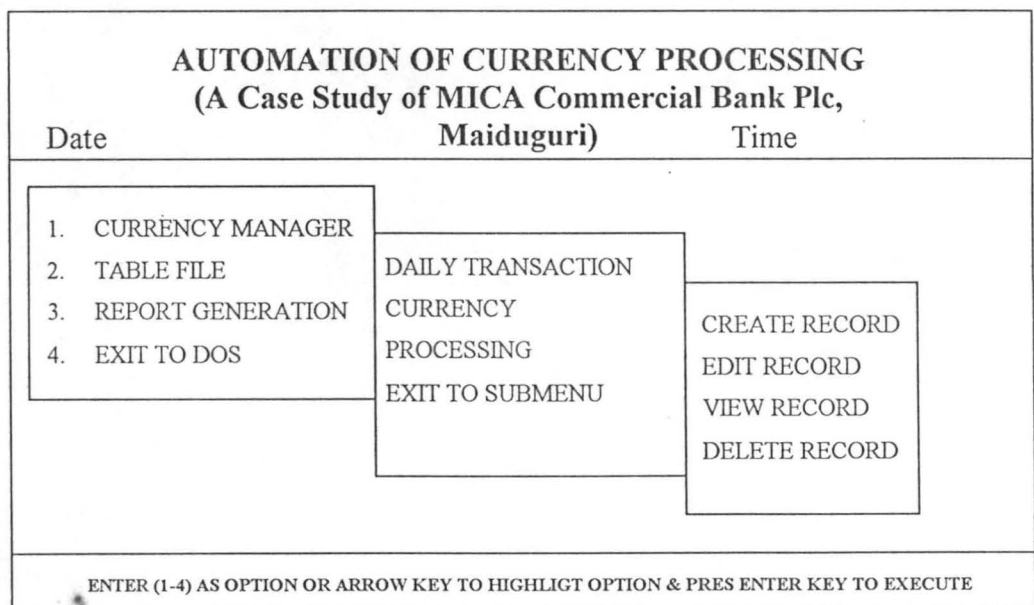


Fig. 2.0 Currency Manager

### 1. ADD RECORD

This is where new records are being entered into the database file concerned. The ADD RECORD is application to both the Currency Manager and the File Option. Moreover, every data entry or transaction made here has a unique code that is used as the key identifier (searching strings) for that particular record.

### 2. EDIT RECORD

The EDIT RECORD allows modification of an existing record in the database file. This is done by entering the unique key identifier called (CODE) to search for the record from the database file and the record is then display on the computer screen to visualize and confirm whether the record display is the one to be modified. If the record displayed is to be modified, a message is prompted at the bottom of the system as shown below:

[M] odify

[R]epeat

[A]bandon

On pressing Letter [M], the system clear the screen and the cursor now blinking to accept new record that will replace the old records in the database file. Pressing letter [R] implies that the process should be repeated for another key identifier (CODE) to be entered in the letter [A] closes the activated database file, clears the system screen and returns back to the main menu system.

### 3 VIEW RECORD

After much data entries the operator might be willing to have a look of data entry he/she has made. The view record option is the best avenue to view or display an existing record in the database file. Here, all the user needs to do is to enter the key identifier (CODE) and the record concerned is displayed on the computer screen.

### 4. DELETE RECORD

All data entries made after a long period of time become obsolete and unwanted in the database file. Such obsolete records are to be deleted from the DELETE RECORD option.

However, it is pertinent to say that any software that inculcates the above four (ADD, EDIT, VIEW AND DELETE) characteristics has met one of the principle conditions of well-structured programs.

\* **DAILY TRANSACTION**

The daily transaction is an avenue where relevant information regarding currency is entered. The information entered here includes: - Transaction date, Operator's code Operator's Name Denomination type and Name, Total Amount Received from customers by operator, Total fit, Unfit and Suspect.

\* **CURRENCY PROCESSING**

This is the second option of the CURRENCY MANAGER. The currency processing is a powerful modules program that processes all daily transaction entries. The module program sorts out banknotes based on their genuineness while counting them. The results of this currency processing are then saved for future use and for further processing of report generation.

**3.4.3 TABLE FILE**

This is the second option of the main - menu system. The table file is used to define certain parameters (information) that are unlikely to change for a long period of time. Such information will include Operator's codes and operator's name, which is under Coding of Operator and Currency Code Submenu.

Note that each of the above submenus has another options (ADD, EDIT, VIEW AND DELETE) as earlier discussed. The general format of the Table is as shown below in Fig. 3.0

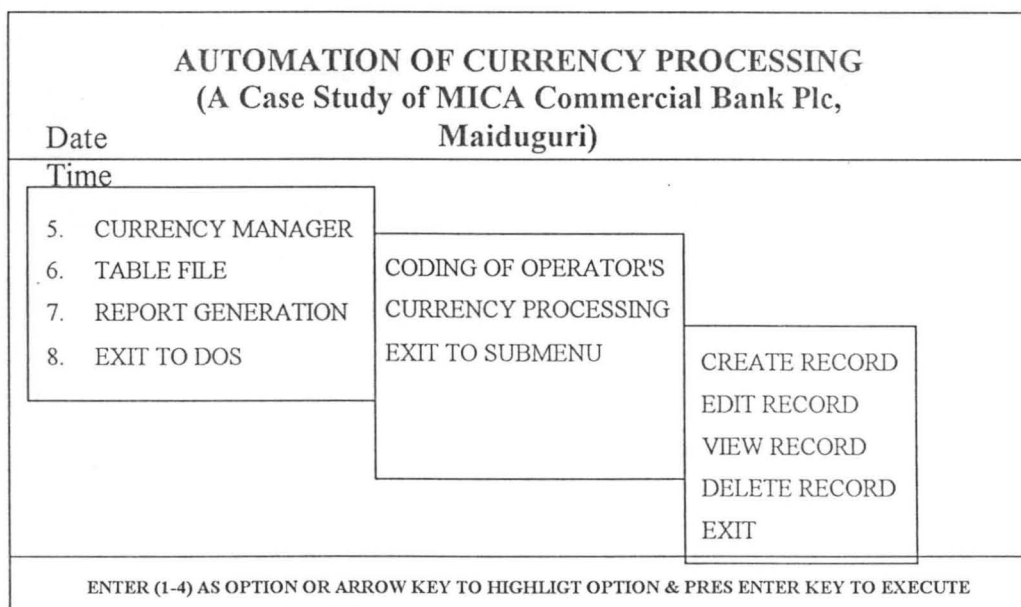


Fig. 3.0 Table File Manager



#### 3.4.4. REPORT GENERATION

This is the their option of the main menu system. This is where relevant reports are generated for the Bank's Management decision making. Few among the expected report are as follows:

- (1) List ACP operators
- (2) List of types of currencies (Denomination)
- (3) Report on total amount received from customers by each operator per day.
- (4) Report on total amount received based on FIT, UNFIT and SUSPECT bank notes.

The report generation in further discussion in section 3.5 of this chapter.

#### 3.5 THE OUTPUT FORMAT

The output format determined the kind of information that goes into the database files. The output format is also known as the output specification shown the detail hard copies produce from the printer. Thus, the ACP was designed to produce two form of output format, they are:

##### \* SCREEN REPORT

In order to save the cost of stationaries (paper), screen report is alternate to production of hard copies, especially, when there are no sufficient papers to be used for report generation, hence the screen report will display detail output of the information on the computer screen for analysis.

##### \* PRINTER REPORT

These are the reports that are directly sent to the printer in order to produce hard copies of information on the paper.

Below are all the available reports being generated from ACP software during the implementation process.

## AUTOMATION OF PROCESSING

(A Case Study of MICA Commercial Bank Plc, Maiduguri)

### REPORT GENERATION FOR LIST OF ACP OPERATORS

---

DATE:

TIME:

---

S/NO	OPERATOR'S CODE	NAME
1	01	LABARAN KAGO
2	02	YUSUF MUHAMMED
3	03	ISAH BALEWA
4	04	HARUNA YOLA
5	05	CHARLES ADAMSON

## AUTOMATION OF PROCESSING

(A Case Study of MICA Commercial Bank Plc, Maiduguri)

### REPORT ON TYPES OF CURRENCIES

---

DATE:

TIME:

---

S/NO	CURRENCY CODE	NAME
1	N1	ONE NAIRA
2	N5	FIVE NAIRA
3	N10	TEN NAIRA
4	N20	TWENTY NAIRA
5	N50	FIFTY NAIRA

## AUTOMATION OF CURRENCY PROCESSING

(A Case Study of MICA Commercial Bank Plc, Maiduguri)

REPORT ON TOTAL AMOUNT RECEIVED BY EACH OPERATOR

DATE:

TIME:

---

S/NO	OPERATOR'S CODENAME	TOTAL AMOUNT
1	01 LABARAN KAGO	500,000.00
2	02 YUSUF MUHAMMED	50,000.00
3	03 HARUNA YOLA	75,000.00
<b>TOTAL CASH DEPOSITED</b>		<b>N635, 000.00</b>

---

## AUTOMATION OF CURRENCY PROCESSING

(A Case Study of MICA Commercial Bank Plc, Maiduguri)

REPORT ON TOTAL AMOUNT RECEIVED BY EACH OPERATOR

DATE:

TIME:

---

S/NO	OP/CODE	NAME	FIT	UNFIT	SUSPECT
1	01	LABARAN KAGO	490,000.00	10,000.00	0.00
2	02	YUSUF MUHAMMED	23,000.00	25,000.00	2,000.00
3	03	HARUNA YOLA	52,000.00	20,000.00	3,000.00
<b>TOTAL FIT, UNFIT &amp; SUSPECT</b>			<b>N565, 000.00</b>	<b>N55, 000.00</b>	<b>N5, 000.00</b>

---

### 3.6 THE INPUT FORMAT

The Input format generally considered to an avenue where the operators and ACP software communicate. This is where all relevant information is being fed into the ACP database file. The input format is also called the Input specification. The general format of the entire available Input format is as shown below: -

AUTOMATED CURRENCY PROCESSING (ACP) (A case Study of Mica Commercial Bank Plc Maiduguri) DATA ENTRY SCREEN FOR OPERATOR'S		
REGISTRATION DATE	<input type="text"/>	
REGISTRATION TIME	<input type="text"/>	
OPERATOR'S CODE	<input type="text"/>	
OPERATOR'S NAME	<input type="text"/>	
(S) AVE	(R) EPEAT	(A) BANDON

AUTOMATED CURRENCY PROCESSING (ACP) (A case Study of Mica Commercial Bank Plc Maiduguri) DATA ENTRY SCREEN FOR REGISTRATION OF CURRENCY	
REGISTRATION DATE	<input type="text"/>
REGISTRATION TIME	<input type="text"/>
CURRENCY CODE	<input type="text"/>
CURRENCY NAME	<input type="text"/>



(2) CURRENCY.DBF

This database file is used to store information regarding registered currency type, e.g. N20 and N10 etc.

S/NO	FILENAME	TYPE	WIDTH	DEC	INDEX
1	C_TYPE	CHR	3	-	-
2	C_NAME	CHR	30	-	-

(3) DAILY.DBF

This database file is used to store all daily transaction made by each operator.

S/NO	FILENAME	TYPE	WIDTH	DEC	INDEX
1	REGDATE	DATE	8	-	-
2	REGTIME	CHR	8	-	-
3	TRANSDATE	DATE	8	-	-
4	USERSCODE	CHR	2	-	-
5	USERSNAME	CHR	30	-	-
6	FITNOTES	NUM	14	-	-
7	UNFITNOTES	NUM	14	-	-
8	FAKENOTES	NUM	14	-	-
9	AMOUNT	NUM	14	-	-

(4) SUMMARY.DBF

This database file is used to store the result of the currency processing that take place. Most report generation are also generated from the information gathered. Below is the structure of the database file.

S/NO	FILENAME	TYPE	WIDTH	DEC	INDEX
1	USERSCODE	CHR	2	-	-
2	USERSNAME	CHR	30	-	-
3	DATE	DATE	8	-	-
4	TOTALFIT	NUM	14	2	-
5	TOTALUNFIT	NUM	14	2	-
6	TOTALFAKE	NUM	14	2	-
7	TOTAL-AMT	NUM	14	2	-

8	C-TYP1	CHR	3	-	-
9	FIT1	NUM	14	2	-
10	UNFIT1	NUM	14	2	-
11	FAKE1	NUM	14	2	-
12	C-TYPE2	CHR	3	-	-
13	FIT2	NUM	14	2	-
14	UNFIT2	NUM	14	2	-
15	FAKE2	NUM	14	2	-
16	C-TYPE3	CHR	3	-	-
17	FIT3	NUM	14	2	-
18	UNFIT3	NUM	14	2	-
19	FAKE3	NUM	14	2	-
20	C-TYP4	CHR	3	-	-
21	FIT4	NUM	14	2	-
22	UNFIT4	NUM	14	2	-
23	FAKE4	NUM	14	2	-
24	C-TYPES	CHR	3	-	-
25	FIT5	NUM	14	2	-
26	UNFIT5	NUM	14	2	-
27	FAKES	NUM	14	2	-
28	C-TYPE6	CHR	3	-	-
29	FIT6	NUM	14	2	-
30	UNFIT6	NUM	14	2	-
31	FAKE6	NUM	14	2	-

### 3.8 FEASIBILITY STUDY

This stage is other wise referred to as preliminary investigation. The feasibility study is embarked upon to determine whether or not the proposed project is desirable. A small group who may be one or two usually carries out this study.

The objectives, which the analysts working on project feasibility should accomplish, include the following:

- (1) Clarify and understand the project Request: -  
The following will aid the attainment of this objective:
  - (a) What is being done?
  - (b) What is required?
  - (c) Why?
- (2) Determine the size of the project. This is necessary, so as to estimate the amount of time and number of people required to develop the project.
- (3) Assess cost and benefit, of the Proposal and the alternative approaches: - What is the cost of the project, including the cost of training and retraining end-users of the information system?
- (4) Determine the Technical, Financial, and Operational Feasibility of alternative approaches.
- (5) Report the findings to management with Recommendation outlining the acceptance or rejection of the proposal.

#### **TESTING PROJECT FEASIBILITY.**

For testing feasibility the following have to be undertaken:

- (a) Operational Feasibility
  - (b) Technical Feasibility
  - (c) Economic Feasibility
- 
- (a) **OPERATIONAL FEASIBILITY:** This relates or is concerned with the working of proposed information system when developed and installed.



# CHAPTER FOUR

## 4.0 IMPLEMENTATION OF THE DESIGNED SOFTWARE

### 4.1 IMPLEMENTATION OF AUTOMATED CURRENCY PROCESSING

The ACP software has been fully designed and developed. The next stage is to implement the system. Implementation is a process that comprises the following: -

- The hardware and software configuration
- Running/Testing and troubleshooting of the ACP and
- The change over procedures

This stage is as important as the overall process of developing the ACP software, this is because, this is where the ACP will be tested with the actual operational data in order to see whether the aims and objectives of developing the ACP has been achieved.

### 4.2 CHOICE OF PROGRAMMING LANGUAGE

#### PROGRAMMING LANGUAGES

Human beings use natural language while computers use artificial languages, it therefore become necessary for human and computer to understand each other and this gives birth to programming languages. The only medium of communicating with computer is by adapting to one or few of its artificial languages.

We can therefore, define programming language as an artificial language designed for use by people in order to ease communication with the computer. Programming languages are divided into *Low Level Language*, *High Level language* and *Program Generator*.

#### (1) LOW LEVEL LANGUAGE

The use of mnemonics or symbol, which are meaningful to the compiler in developing instructions, which the computer will follow in achieving a given task is called Low level Language. An example of low level language is Assembly Language.

#### (2) HIGH LEVEL LANGUAGE

These are computer languages, which are more like English words, phrase and vocabularies. Example of the language includes Ada Language, Prolog1, Pascal Basic, Cobol, FORTRAN, etc.

### 3.8 FEASIBILITY STUDY

This stage is other wise referred to as preliminary investigation. The feasibility study is embarked upon to determine whether or not the proposed project is desirable. A small group who may be one or two usually carries out this study.

The objectives, which the analysts working on project feasibility should accomplish, include the following:

- (1) Clarify and understand the project Request: -  
The following will aid the attainment of this objective:
  - (a) What is being done?
  - (b) What is required?
  - (c) Why?
- (2) Determine the size of the project. This is necessary, so as to estimate the amount of time and number of people required to develop the project.
- (3) Assess cost and benefit, of the Proposal and the alternative approaches: - What is the cost of the project, including the cost of training and retraining end-users of the information system?
- (4) Determine the Technical, Financial, and Operational Feasibility of alternative approaches.
- (5) Report the findings to management with Recommendation outlining the acceptance or rejection of the proposal.

#### **TESTING PROJECT FEASIBILITY.**

For testing feasibility the following have to be undertaken:

- (a) Operational Feasibility
  - (b) Technical Feasibility
  - (c) Economic Feasibility
- 
- (a) **OPERATIONAL FEASIBILITY:** This relates or is concerned with the working of proposed information system when developed and installed.

- (b) **TECHNICAL FEASIBILITY:** This test seeks to clarify if the proposed project can be done with current equipment, existing software technology and available personnel.
- (c) **ECONOMICAL FEASIBILITY:** The tests for financial feasibility is undertaken to assess cost of implementing a proposed project vis-a-vis benefit derived in implementing the project.

### **REQUIREMENT DETERMINATION**

A requirement is a feature that must be included in a new system. The requirement determination entails studying the existing system and collection of detailed information about the system so as to find out what their requirements are.

Requirement Determination consist of 3 activities namely:

- (a) Requirement Anticipation (RA)
  - (b) Requirement Investigation (RI)
  - (c) Requirement Specification (RS)
- 
- (a) **Requirement Anticipation (RA):** - These are problems or features and requirements the analyst facing a new system may have as a result of his previous experience.
  - (b) **Requirement Investigation (RI):** - This activity uses a variety of tools and skills to study current system ,document and its features for further analysis.
  - (c) **Requirement Specification (RS):** - This activity leads to a description of features for new system based on the analyst of data produced during the fact - finding investigation. Requirement specification should be thoroughly done for the quality of the performance at this point would reflect later in the characteristic of the new evolved.

### **3.9 SYSTEM COVERSION/CHANGE OVER PROCEDURES**

This is the last stage of implementation. This stage involves changing from the old system (manual - based method) to the newly designed software (ACP software).

However, in changing from old method to the newly designed system, four methods were looked into, these methods include the following: -

- a) Parallel change over procedure
- b) Stage change over procedure
- c) Direct change over procedure
- d) Pilot change over procedure

(a) **PARALLEL CHANGE OVER PROCEDURE**

This process involves parallel running of the two (2) system, that is both the manual method and the new system all together by comparing the result of the old and new system until the new system proves efficient and excellent before the running of the old system is terminated

(b) **STAGED CHANGE OVER PROCEDURE**

This method of changeover procedure involves stage by stage. This is based on the computer procedural program (Modules). Here, a module of the program must be completed, tested and found to be efficient before proceeding to the next stage of the program. In no circumstance the next stage will be reached without accomplishing and ensuring that the first stage is absolutely okay.

(c) **DIRECT CHANGE OVER PROCEDURE**

This process involves changing from the old system to the newly designed system directly. Thus, it is expected that the system must have been tested and found efficient before the direct changeover can take place.

(d) **PILOT CHANGE OVER PROCEDURE**

This is similar in concept to parallel running. Data from one or more previous periods for the whole or part of the system is run on the new system after result have been obtained from the old system, and the new results are compared with the old. It is not as disruptive as parallel operation, since timing is less critical. This method is more like an extended system test, but it may be considered a more practical form of change over for organizational reasons.

Going by the above analysis, Parallel Change over procedure was used in order to actualized the aims and objectives ACP software proved to be accurate and efficient, with increase in speed of processing, the old system was then phased out and the ACP Software becomes fully operational

### 3.10 COST AND BENEFIT ANALYSIS

The Cost Benefit Analysis, the analysis of the total cost (expenses) that is needed in order to put the ACP Software into actual implementation irrespective of environment, people and other constraint that may deter the implementation.

However, it is quite important to note that, cost benefit analysis might be is difficult to quantify at best, but must be done in order to estimate the financial and operational impact it will have on the organization concerned.

Below is the cost benefit analysis of implementing the ACP Software

DESCRIPTION	UNIT	RATE	AMONUT	
S/NO				
<b>(A). OPERATIONAL COST</b>				
1.	Complete Computer System	1	200,000	200,000.00
2.	HP Laser Jet 6L Printer	1	65,000	65,000.00
3.	Epson LQ 2170 Printer	1	95,000	95,000.00
4.	Stabilizer (AVR)	1	55,000	55,000.00
5.	UPS (Un-Interrupted Power Supply)	1	65,000	65,000.00
6.	Equipment Maintenance	Lot	70,000	70,000.00
7.	Miscellaneous Expenses	Lot	20,000	20,000.00
<b>TOTAL ON ITEM (A)</b>				<b>570,000.00</b>
<b>(B) DEVELOPMENTAL COST</b>				
1.	System Analysis & Design, Program	Lot	150,000	150,000.00
2.	Development	Lot	50,000	50,000.00
3.	ACP Software Implementation	staff	10,000	10,000.00
4.	Personnel Training for One Month	lot	25,000	25,000.00
ACP Installation				
<b>TOTAL ON ITEM (B)</b>				<b>275,000.00</b>
<b>GRAND TOTAL ON ITEM (A) &amp; (B)</b>				<b>845,000.00</b>

A total of Eight Hundred and Forty Five Thousand Naira (N845, 000.00) will be needed to actually implement the ACP Software.

The benefit to be derived from the implementation of the ACP Software will include the followings:

1. Timely presentation of Currency Processed report
2. Eliminate duplication of currency record as experienced in the manual method
3. Adequate sorting of currency while counting
4. ACP Software is developed to be Year 2000 Compliant
5. Reduce redundancy an increases efficiency of currency processing operation

### **(3) PROGRAM GENERATOR LANGUAGE**

Program Generator is an application, which provides the user with the tool for designing business applications. Program generator is capable of constructing set of instructions (Program) as you select already-made options. These instructions (program) can be compiled into executable file.

Examples of this language are: - (1) Database Management system (DBMS) packages. (Dbase IV, clipper 2.0, FoxBASE, SQL) (2) Object Oriented Programming Language (Visual Basic, Java, and Oracle) etc.

Going by the above analysis of programming languages, the choice of the programming language for the Automate currency processing (ACP) is the program Generator, which is considered to be a tool for designing business application package.

However, the choice under this group of language is the Database Management System package (i.e. Dbase IV and clipper 2.0) because of the following facts: -

- (1) Dbase iv and clipper 2.0 are user's friendly
- (2) They are both easy to use and understand
- (3) They provide Active user interface program
- (4) Clipper 2.0 posses a translator called the compiler that enables source program to be compiled to executable file. Hence, the ACP can be run independent of the application packages that were used in developing it.
- (5) The two application packages provide flow of data during data entry.
- (6) The two Database Management packages allow the storage of large volume record in the database file.
- (7) They provide fast access to large volume of records in the database file.

### **4.3 HARDWARE AND SOFTWARE CONFIGURATION REQUIREMENT**

Configuring a computer means setting up a computer so that Disk operating system (DOS) windows, Hardware and Software can effectively function and maximize the use of the available physical and logical resources.

The configuration of the Hardware and software requirements for the running of the ACP maximizes the utilization of the computer through the use of the software that was installed.

## **HARDWARE FEATURES REQUIREMENT**

This includes the minimum physical features of the system needed for the running of the developed ACP software and other application software. The following minimum features for proper implementation is required.

- PENTIUM 300MHZ Celeron Intel
- 40 GB HDD
- 32 MB RAM
- 36 X BIT CD-ROM Drive+(2) 100 Watt Speakers
- 14" SVGA MONITOR ENHANCED KEYBOARD
- UPS (UN-INTERRUPTED POWER SUPPLY)
- STABILIZER (AUTOMATIC VOLTAGE REGULATOR)

## **SOFTWARE REQUIREMENT**

The software requirement includes all the software that will be needed for the running of ACP and other application software.

### **OPERATING SYSTEM SOFTWARE**

- (1) MS-DOS
  - (2) WINDOWS 95 OR 98
  - (3) ACP SOFTWARE
  - (4) OTHER APPLICATION SOFTWARE
- 
- (a) MICROSOFT OFFICE 97
  - (b) WORD PERFECT 6.1 FOR WINDOWS
  - (c) PAGE MAKER
  - (d) COREL DRAW
  - (e) DBASE (4)
  - (f) CLIPPER 2.0

## **4.4 THE ACP INSTALLATION**

ACP installation is an important part of the implementation. The idea is that the developed system should be installed on the above mentioned computer system in order to see and



check whether the aims are achieved. Below are procedures to be followed for proper installation of the ACP software:

STAGES	DESCRIPTION	FORMAT
(1)	Create a Directory in a Drive C named as {ACP}	C:\>MD ACP (Enter Key)
(2)	Change from the Root Directory to ACP	C:\>MD ACP (Enter Key)
(3)	ACP now becomes the default directory	C:\>MD ACP
(4)	Insert the ACP first Installation diskettes In Drive A, and Change default Drive to A.	C:\>MD ACP (Enter Key)
(5)	Using the COPY COMMAND Copy all the files Contained in Drive A to C	A:\>MD *. * C: (Enter Key)
(6)	Change Default drive back to drive C	A:\>C: (Enter Key)
(7)	ACP is now the default Drive, type ACP to run the ACP software	C:\ACP> ACP
(8)	Displays the below messages <i>"Welcome To ACP Software</i>	

## **4.5 TEST RUNNING/TROUBLE SHOOTING**

This stage involves trouble shooting/(test running) the software and ensuring that it is error-free. In the process of trouble shooting all the logical, syntax and unrecognised error encountered have been fixed. Hence, the ACP has been tested with a real life data and was found to be efficient.

## **4.6 STARTING THE NEW SYSTEM**

To start the Automated Currency processing (ACP) Software is very simple and entails the following procedures"

1. At MS-Dos Prompt, change directory name to ACP
2. Type ACP and press the ENTER key to start the New system.

# CHAPTER FIVE

## 5.0 SUMMARY AND CONCLUSION

### 5.1 SUMMARY

The project Automation of currency processing is broken down (summarized) into five (5) chapters with the ultimate aims of transforming the manual base of currency processing into an Automated currency processing. Each chapter is unified with a chapter heading as below.

Chapter one (Introduction to currency processing) discussed, the introduction, Links between computer and the Counting Machine, brief history of MICA Commercial Bank Plc, Maiduguri, project motivation, objectives of the study, research methodology, the basic concept and scope of the study.

Chapter two (Literature Review) were analysed, starting with Functional Description of Currency Processing, the currency Management, the Existing Operation at the Bank, numerous problems associated with the existing operation. Consequently, possible solutions to these numerous problems were recommended. The advantages of currency processing were discussed, limitation of the manual based currency processing and to end chapter, the basic features of automated Currency processing were listed and summarized.

However, chapter three (System Analysis and design) gives an in-depth explanation on System Analysis, Design and development Stages. Analysis was also carried out on the followings: -

- The proposed automated currency processing (ACP),
- The limitation of the automated currency processing,
- The structure of the new system,
- The output and input format,
- The database file (file design) and
- The choice of the programming language used in developing the ACP software.

Emphasizes were made on chapter four (implementation of the designed software). This includes the implementation of ACP, the Hardware and Software configuration Requirement,

and the Installation process for the ACP, test Running and trouble shooting of the ACP and lastly change over procedure involved in changing the newly designed system (ACP).

Lastly, Chapter five (5), Summary and Condition give a comprehension Summary of the whole project work into chapter.

## **5.2 CONCLUSION**

It can be conclusively said that, the ACP software has been duly tested with real life data, under human and different environmental changes and the out come is that, it has proved to be efficient, increase in speed of accessibility and reliable with minimum human resources. Hence, it is suffix to say that a computer gives solution to virtually all problems that can be well defined. The solution to the problems depend on the following features:

- (1). The Computer Hardware and Software Configuration
- (2). The task (job) placed on the Computer System and
- (3). The ability to make maximum utilization of the system.

## **5.3 RECOMMENDATIONS**

Going by the increase in the numbers of customers of the bank and in order to be able to handle the currency flowing back from circulation with ease in accordance with the Central Bank of Nigeria requirement in currency processing, it is pertinent to recommend the following.

- The ACP software should sooner than later be networked in order to further increase the speed of currency processing at the MICA Commercial Bank Plc, Maiduguri.
- Thus, the ACP software is only limited to MICA Commercial Bank Plc, Maiduguri, it is still subject to modification and can be recommended to other commercial banks in the country.

## REFERENCES

- 1: Akin Fapounda (1996) - Understanding Microcomputers  
Published by Aflon Publication, Abuja
2. Akin Fapounda (1998) - Introductory to Computer Science  
Published by Aflon Publication, Abuja
3. Badamasi Abdullrasheed (1996)- System Analysis and Design  
Course Material (CPD 728), FUT Minna  
(Unpublished)
4. Giesecke and Devrient (1988) - Introductory to International Security  
Systems  
Published by Giesecke and Deviant,  
Germany.
5. Kola Raimi (1997) - Understanding Database Management  
System.  
Course Material (CPD 726), FUT Minna,  
(Unpublished).

```

*****
***** NAME : CHARLES EJIKE ANYAEGBU *****
*****
***** NUMBER : PGD/MCS/96/100 *****
*****
***** TOPIC : AUTOMATION OF CURRENCY PROCESSING *****
*****

```

```

Set Talk off
Set Echo off
Set Bell off
Set Stat off
Set Scor off
Set Safe off
Set Conf off
Set Esca off
Set Wrap on
Set date to brit
private sc_hpop // Saves horizontal popup
clear
x=inkey(2)
set color to w+/br, ,
@00,00 clear to 05,79
@00,00 to 05,79 color "w+/r"
cbn1="AUTOMATED CURRENCY PROCESSING (ACP)"
cbn2="(A case Study of MICA Commercial Bank Plc. Maiduguri)"
@01,(80-len(cbn1))/2 say cbn1
@02,(80-len(cbn2))/2 say cbn2

```

```

private sc_vpop
Do while .t.
Set color to w+/b
@06,00 clear to 14,38
@06,00 to 14,38

@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use'+chr(25)+'or'+chr(24)+' key to highlight option &
press '+chr(17)+chr(196)+chr(217)+' key to select'
@07,5 prompt " CURRENCY MANAGER"
@09,5 prompt " TABLE FILE"
@11,5 prompt " REPORT GENERATION"
@13,5 prompt " EXIT TO DOS"
Save screen to sc_vpop
menu to fchoice
Do case
    case fchoice=0

```

```
    loop
  case fchoice=1
    do CURRENCY
  case fchoice=2
    do TABLE
  case fchoice=3
    do REPORT
  othe
  exit
Endcase
Restore Screen from sc_hpop
Enddo
Return
```

Procedure CURRENCY

```
private sc2_vpop
```

```
Do while .t.
```

```
Set color to w+/b
```

```
@08,30 clear to 18,57
```

```
@08,30 to 18,57
```

```
@21,08 clear to 23,71
```

```
@21,08 to 23,71
```

```
@22,10 say 'Use'+chr(25)+'or'+chr(24)+' key to highlight option &
press '+chr(17)+chr(196)+chr(217)+' key to select'
```

```
@09,32 prompt " DAILY TRANSACTION"
```

```
@11,32 prompt " CURRENCY"
```

```
@13,32 prompt " OPERATORS"
```

```
@15,32 prompt " EXIT TO SUBMENU"
```

```
Save screen to sv2_vpop
```

```
menu to ch2
```

```
Do case
```

```
  case ch2=0
```

```
    loop
```

```
  case ch2=1
```

```
    do DAILYMAIN
```

```
  case ch2=2
```

```
    do CURREMAIN
```

```
  case ch2=3
```

```
    do PROCMAIN
```

```
  othe
```

```
    exit
```

```
Endcase
```

```
Restore Screen from sc_vpop
```

```
Enddo
```

```
Return
```

```

Procedure TABLE
private sc2_vpop
Do while .t.
Set color to w+/b
@08,30 clear to 18,57
@08,30 to 18,57

@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use'+chr(25)+'or'+chr(24)+' key to highlight option &
press '+chr(17)+chr(196)+chr(217)+' key to select'
@09,32 prompt " CODING OF OPERATORS"
@11,32 prompt " CURRENCY PROCESSING"
@13,32 prompt " EXIT TO SUBMENU"

Save screen to sv2_vpop
menu to ch2
Do case
  case ch2=0
    loop
  case ch2=1
    *do DAILY
    EXIT
  case ch2=2
    *do CURRENT
    EXIT
  othe
  exit
Endcase
Restore Screen from sc_vpop
Enddo
Return

```

```

Procedure DAILYMAIN
private sc3_vpop
Do while .t.
Set color to w+/b
@08,49 clear to 18,67
@08,49 to 18,67

@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use'+chr(25)+'or'+chr(24)+' key to highlight option &
press '+chr(17)+chr(196)+chr(217)+' key to select'
@09,50 prompt " Create Record"
@11,50 prompt " Modify Record"

```



```
@13,50 prompt " View Record"  
@15,50 prompt " Delete Record"  
@17,50 prompt " Exit Menu"
```

```
Save screen to sv3_pop
```

```
menu to ch2
```

```
Do case
```

```
  case ch2=0
```

```
    loop
```

```
  case ch2=1
```

```
    do DAILYADD
```

```
  case ch2=2
```

```
    do DAILYMOD
```

```
  case ch2=3
```

```
    do DAILYVIEW
```

```
  case ch2=4
```

```
    do DAILYDEL
```

```
  othe
```

```
    exit
```

```
Endcase
```

```
Restore Screen from sc2_vpop
```

```
Enddo
```

```
Return
```

```
Procedure CURREMAIN
```

```
private sc3_vpop
```

```
Do while .t.
```

```
Set color to w+/b
```

```
@08,49 clear to 18,67
```

```
@08,49 to 18,67
```

```
@21,08 clear to 23,71
```

```
@21,08 to 23,71
```

```
@22,10 say 'Use'+chr(25)+'or'+chr(24)+' key to highligt option &  
press '+chr(17)+chr(196)+chr(217)+' key to select'
```

```
@09,50 prompt " Create Record"
```

```
@11,50 prompt " Modify Record"
```

```
@13,50 prompt " View Record"
```

```
@15,50 prompt " Delete Record"
```

```
@17,50 prompt " Exit Menu"
```

```
Save screen to sv3_vpop
```

```
menu to ch2
```

```
Do case
```

```
  case ch2=0
```

```
    loop
```

```
  case ch2=1
```

```
do CURREADD
case ch2=2
do CURREMOD
case ch2=3
do CURREVIEW
case ch2=4
do CURREDEL
othe
exit
Endcase
Restore Screen from sc2_vpop
Enddo
Return
```

```
Procedure PROCMAIN
private sv3_vpop
Do while .t.
Set color to w+/b
@08,49 clear to 18,67
@08,49 to 18,67

@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use'+chr(25)+'or'+chr(24)+' key to highlight option &
press '+chr(17)+chr(196)+chr(217)+' key to select'
@09,50 prompt " Create Record"
@11,50 prompt " Modify Record"
@13,50 prompt " View Record"
@15,50 prompt " Delete Record"
@17,50 prompt " Exit Menu"
```

```
Save screen to sv3_vpop
menu to ch2
Do case
case ch2=0
loop
case ch2=1
do PROCADD
case ch2=2
do PROCMOD
case ch2=3
do PROCVIEW
case ch2=4
do PROCDEL
othe
exit
Endcase
```

```
Restore Screen from sc2_vpop
Enddo
Return
```

```
*/-----Menu Choice for Report Program-----
Procedure report
private sc_vpop
Do while .t.
Set color to w+/b
@06,19 clear to 18,75
@06,19 to 18,75

@21,08 clear to 23,71
@21,08 to 23,71
@22,10 say 'Use'+chr(25)+'or'+chr(24)+' key to highlight option &
press '+chr(17)+chr(196)+chr(217)+' key to select'
@07,20 prompt " 1. List ACP Operators..."
@09,20 prompt " 2. List of types of Currencies
(Denomination)....."
@11,20 prompt " 3. Total Amount received by each
operator....."
@13,20 prompt " 4. Total Amount received based on FIT UNFIT &
SUSPECT."
@15,20 prompt " 6.Exit Main
Menu....."
```

```
Save screen to sc_vpop
menu to fchoice
Do case
  case fchoice=0
  loop
  case fchoice=1
  do rep1
  case fchoice=2
  do rep2
  case fchoice=3
  do rep3
  case fchoice=4
  do rep4
  othe
  exit
Endcase
Restore Screen from sc_hpop
Enddo
Return
```

```

Procedure DAILYADD
clear
set talk off
set status off
store space(30) to mUsername
store space(2) to muserscode
store space(8) to mregtime
store ctod(" / / ") to mregdate,mtransdate
store 0.00 to mfitnotes, munfitnotes, mfakenotes, mamount
use daily
do while .t.
  store "Y" to yn
  @ 2,27 SAY "Data Entry Form At Front Desk" color 'rg+/n'
  @ 4,6 TO 20,70 DOUBLE color 'rg'
  @ 6,10 SAY "REGISTRATION DATE :"
  @ 6,30 GET mregdate
  read
  locate all for mregdate = regdate
  if found()
    @ 8,10 say "Record Already Exist ...."
    wait ""
  else
    @ 8,10 SAY "REGISTRATION TIME :"
    @ 8,30 GET mregtime pict "99:99:99"
    @ 10,10 SAY "OPERATOR'S CODE   :"
    @ 10,30 GET muserscode pict "XX"
    @ 12,10 SAY "OPERATOR'S NAME   :"
    @ 12,30 GET musername pict "@!"
    @ 14,10 SAY "TOTAL NO. OF FIT BANK NOTES RECEIVED :"
    @ 14,50 GET mfitnotes pict "9999999.99"
    @ 16,10 SAY "TOTAL NO. OF UNFIT BANK NOTES RECEIVED :"
    @ 16,50 GET munfitnotes pict "9999999.99"
    @ 18,10 SAY "TOTAL NO. OF SUSPECT BANK NOTES RECEIVED :"
    @ 18,50 GET mfakenotes pict "9999999.99"
  read
  @ 19,20 say "Are These Entries OK(Y/N) ?"
  @ 19,48 get yn pict "!"
  read
  if upper(yn) $'N'
    loop
  else
    if upper(yn)$"Y"
      append blank
      replace username with musername, userscode with
muserscode
      replace regtime with mregtime,regdate with mregdate
      replace transdate with mtransdate

```

```

        replace fitnotes with mfitnotes, unfitnotes with
munfitnotes
        replace fakenotes with mfakenotes
        mamount = mfitnotes + mfakenotes
        replace amount with mamount
    endif
endif
endif
yn='Y'
@ 19,20 clear to 19,60
@ 19,20 say "Add more Records (Y/N) ?"
@ 19,43 get yn pict "!"
read
if upper(yn) $ 'Y'
    clear
        store space(30) to mUsername
        store space(2) to muserscode
        store space(8) to mregtime
        store ctod(" / / ") to mregdate,mtransdate
        store 0.00 to mfitnotes, munfitnotes, mfakenotes, mamount
    loop
else
    if upper(yn) $ 'N'
        exit
    endif
endif
endif
enddo
clear
close data
return

```

#### Procedure DAILYDEL

```

clear
set talk off
set status off
store space(30) to mUsername
store space(2) to muserscode
store space(8) to mregtime
store ctod(" / / ") to mregdate,mtransdate
store 0.00 to mfitnotes, munfitnotes, mfakenotes, mamount
use daily
do while .t.
    store "Y" to yn
    @ 2,27 SAY "Delete records from File" color 'rg+/n'
    @ 4,6 TO 20,70 DOUBLE color 'rg'
    @ 6,10 SAY "REGISTRATION DATE : "
    @ 6,30 GET mregdate

```

```

read
locate all for mregdate = regdate
if found()
clear
store "Y" to yn
@ 8,10 SAY "REGISTRATION TIME : "
@ 8,30 GET mregtime pict "99:99:99"
@ 10,10 SAY "OPERATOR'S CODE   : "
@ 10,30 GET muserscode pict "XX"
@ 12,10 SAY "OPERATOR'S NAME   : "
@ 12,30 GET musersname pict "@!"
@ 14,10 SAY "TOTAL NO. OF FIT BANK NOTES RECEIVED : "
@ 14,50 GET mfitnotes pict "9999999.99"
@ 16,10 SAY "TOTAL NO. OF UNFIT BANK NOTES RECEIVED : "
@ 16,50 GET munfitnotes pict "9999999.99"
@ 18,10 SAY "TOTAL NO. OF SUSPECT BANK NOTES RECEIVED : "
@ 18,50 GET mfakenotes pict "9999999.99"
@ 19,20 say "Delete this record(Y/N) ?"
@ 19,48 get yn pict "!"
read
if upper(yn) $ 'Y'
    delete
    pack
else
    exit
endif
else
    if .not. found()
        clear
        @ 7,10 Say "No Such Record Exist In Database, Press any key
to Continue..."
        set console off
        wait
        set console off
    endif
endif
enddo
clear
close data
return

```

Procedure DAILYMOD

```

clear
set talk off
set status off
store space(30) to mUsername
store space(2) to muserscode

```

```

store space(8) to mregtime
store ctod(" / / ") to mregdate,mtransdate
store 0.00 to mfitnotes, munfitnotes, mfakenotes, mamount
use daily
do while .t.
  store "Y" to yn
  @ 2,27 SAY "Modification of Records" color 'rg+/n'
  @ 4,6 TO 20,70 DOUBLE color 'rg'
  @ 6,10 SAY "REGISTRATION DATE :"
  @ 6,30 GET mregdate
  read
  locate all for mregdate = regdate
  if .not. found()
    @ 8,10 say "Record Does not Exist ...."
    wait ""
  else
    store username to mUsername
    store userscode to muserscode
    store regtime to mregtime
    store regdate to mregdate
    store transdate to mtransdate
    store fitnotes to mfitnotes
    store unfitnotes to munfitnotes
    store fakenotes to mfakenotes
    store amount to mamount
  @ 8,10 SAY "REGISTRATION TIME :"
  @ 8,30 GET mregtime pict "99:99:99"
  @ 10,10 SAY "OPERATOR'S CODE   :"
  @ 10,30 GET muserscode pict "XX"
  @ 12,10 SAY "OPERATOR'S NAME   :"
  @ 12,30 GET musername pict "@!"
  @ 14,10 SAY "TOTAL NO. OF FIT BANK NOTES RECEIVED :"
  @ 14,50 GET mfitnotes pict "9999999.99"
  @ 16,10 SAY "TOTAL NO. OF UNFIT BANK NOTES RECEIVED :"
  @ 16,50 GET munfitnotes pict "9999999.99"
  @ 18,10 SAY "TOTAL NO. OF SUSPECT BANK NOTES RECEIVED :"
  @ 18,50 GET mfakenotes pict "9999999.99"
  read
  @ 19,20 say "Are These Entries OK(Y/N) ?"
  @ 19,48 get yn pict "!"
  read
  if upper(yn) $'N'
    loop
  else
    if upper(yn)$"Y"
      replace username with musername, userscode with
muserscode

```

```

        replace regtime with mregtime, regdate with mregdate
        replace transdate with mtransdate
        replace fitnotes with mfitnotes, unfitnotes with
munfitnotes
        replace fakenotes with mfakenotes
        mamount = mfitnotes + mfakenotes
        replace amount with mamount
    endif
endif
endif
yn='Y'
@ 19,20 clear to 19,60
@ 19,20 say "Modify more Records (Y/N) ?"
@ 19,43 get yn pict "!"
read
if upper(yn) $ 'Y'
    clear
        store space(30) to mUsername .
        store space(2) to muserscode
        store space(8) to mregtime
        store ctod(" / / ") to mregdate,mtransdate
        store 0.00 to mfitnotes, munfitnotes, mfakenotes, mamount
    loop
else
    if upper(yn) $ 'N'
        exit
    endif
endif
enddo
clear
close data
return

```

#### Procedure DAILYVIEW

```

clear
set talk off
set status off
store space(30) to mUsername
store space(2) to muserscode
store space(8) to mregtime
store ctod(" / / ") to mregdate,mtransdate
store 0.00 to mfitnotes, munfitnotes, mfakenotes, mamount
use daily
do while .t.
    store "Y" to yn
    @ 2,35 SAY "Viewing Records" color 'rg+/n'

```



```

@ 4,6 TO 20,70 DOUBLE color 'rg'
@ 6,10 SAY "REGISTRATION DATE :"
@ 6,30 GET mregdate
read
locate all for mregdate = regdate
if .not. found()
    @ 8,10 say "Record Does not Exist ...."
    wait ""
else
    store username to mUsername
    store userscode to muserscode
    store regtime to mregtime
    store regdate to mregdate
    store transdate to mtransdate
    store fitnotes to mfitnotes
    store unfitnotes to munfitnotes
    store fakenotes to mfakenotes
    store amount to mamount
@ 8,10 SAY "REGISTRATION TIME :"
@ 8,30 GET mregtime pict "99:99:99"
@ 10,10 SAY "OPERATOR'S CODE   :"
@ 10,30 GET muserscode pict "XX"
@ 12,10 SAY "OPERATOR'S NAME   :"
@ 12,30 GET musersname pict "@!"
@ 14,10 SAY "TOTAL NO. OF FIT BANK NOTES RECEIVED :"
@ 14,50 GET mfitnotes pict "9999999.99"
@ 16,10 SAY "TOTAL NO. OF UNFIT BANK NOTES RECEIVED :"
@ 16,50 GET munfitnotes pict "9999999.99"
@ 18,10 SAY "TOTAL NO. OF SUSPECT BANK NOTES RECEIVED :"
@ 18,50 GET mfakenotes pict "9999999.99"
read
wait ""
loop
endif
yn='Y'
@ 19,20 clear to 19,60
@ 19,20 say "View more Records (Y/N) ?"
@ 19,43 get yn pict "!"
read
if upper(yn) $ 'Y'
    clear
    store space(30) to mUsername
    store space(2) to muserscode
    store space(8) to mregtime
    store ctod(" / / ") to mregdate,mtransdate
    store 0.00 to mfitnotes, munfitnotes, mfakenotes, mamount
loop

```

```
else
  if upper(yn) $'N'
    exit
  endif
endif
enddo
clear
close data
return
```

Procedure CURREADD

```
clear
set talk off
set status off
store space(30) to mc_name
store space(3) to mc_type
use currency
do while .t.
  store "Y" to yn
  @ 2,27 SAY "Data Entry Form At Front Desk" color 'rg+/n'
  @ 4,6 TO 20,70 DOUBLE color 'rg'
  @ 6,10 SAY "CURRENCY CODE      :"
  @ 6,30 GET mc_type pict "XXX"
  read
  locate all for mc_type = c_type
  if found()
    @ 8,10 say "Record Already Exist ...."
    wait ""
  else
    @ 8,10 SAY "CURRENCY NAME      :"
    @ 8,30 GET mc_name pict "@!"
    read
    @ 19,20 say "Are These Entries OK(Y/N) ?"
    @ 19,48 get yn pict "!"
    read
    if upper(yn) $'N'
      loop
    else
      if upper(yn)$"Y"
        append blank
        replace c_type with mc_type, c_name with mc_name
      endif
    endif
  endif
  yn='Y'
  @ 19,20 clear to 19,60
  @ 19,20 say "Add more Records (Y/N) ?"
```

```

@ 19,43 get yn pict "!"
read
if upper(yn) $ 'Y'
  clear
  store space(30) to mc_name
  store space(3) to mc_type
  loop
else
  if upper(yn) $ 'N'
    exit
  endif
endif
enddo
clear
return

```

Procedure CURREDEL

```

clear
set talk off
set status off
store space(30) to mc_name
store space(3) to mc_type
use currency
do while .t.
  store "Y" to yn
  @ 2,27 SAY "Delete records from File" color 'rg+/n'
  @ 4,6 TO 20,70 DOUBLE color 'rg'
  @ 6,10 SAY "CURRENCY CODE      :"
  @ 6,30 GET mc_type pict "XXX"
  read
  locate all for mc_type = c_type
  if found()
    clear
    store "Y" to yn
    @ 8,10 SAY "CURRENCY NAME      :"
    @ 8,30 GET mc_name pict "@!"
    @ 19,20 say "Delete this record(Y/N) ?"
    @ 19,48 get yn pict "!"
    read
    if upper(yn) $ 'Y'
      delete
      pack
    else
      exit
    endif
  else
    if .not. found()

```

```

clear
@ 7,10 Say "No Such Record Exist In Database, Press any key
to Continue..."
set console off
wait
set console off
endif
endif
enddo
clear
return

```

#### Procedure CURREMOD

```

clear
set talk off
set status off
store space(30) to mc_name
store space(3) to mc_type
use currency
do while .t.
store "Y" to yn
@ 2,27 SAY "Modification of Records" color 'rg+/n'
@ 4,6 TO 20,70 DOUBLE color 'rg'
@ 6,10 SAY "CURRENCY CODE      :"
@ 6,30 GET mc_type pict "XXX"
read
locate all for mc_type = c_type
if .not. found()
@ 8,10 say "Record Does not Exist ...."
wait ""
else
store c_type to mc_type
store c_name to mc_name
@ 8,10 SAY "CURRENCY NAME      :"
@ 8,30 GET mc_name pict "@!"
read
@ 19,20 say "Are These Entries OK(Y/N) ?"
@ 19,48 get yn pict "!"
read
if upper(yn) $'N'
loop
else
if upper(yn) $"Y"
replace c_type with mc_type, c_name with mc_name
endif
endif
endif
endif

```

```

yn='Y'
@ 19,20 clear to 19,60
@ 19,20 say "Add more Records (Y/N) ?"
@ 19,43 get yn pict "!"
read
if upper(yn) $ 'Y'
    clear
    store space(30) to mc_name
    store space(3) to mc_type
    loop
else
    if upper(yn) $ 'N'
        exit
    endif
endif
enddo
clear
close data
return

```

Procedure CURREVIEW

```

clear
set talk off
set status off
store space(30) to mc_name
store space(3) to mc_type
use currency
do while .t.
    store "Y" to yn
    @ 2,35 SAY "Viewing Records" color 'rg+/n'
    @ 4,6 TO 20,70 DOUBLE color 'rg'
    @ 6,10 SAY "CURRENCY CODE      :"
    @ 6,30 GET mc_type pict "XXX"
    read
    locate all for mc_type = c_type
    if .not. found()
        @ 8,10 say "Record Does not Exist ...."
        wait ""
    else
        store c_type to mc_type
        store c_name to mc_name
    @ 8,10 SAY "CURRENCY NAME      :"
    @ 8,30 GET mc_name pict "@!"
    read
    wait ""
    loop

```

```

endif
yn='Y'
@ 19,20 clear to 19,60
@ 19,20 say "Add more Records (Y/N) ?"
@ 19,48 get yn pict "!"
read
if upper(yn) $ 'Y'
  clear
  store space(30) to mc_name
  store space(3) to mc_type
  loop
else
  if upper(yn) $ 'N'
    exit
  endif
endif
enddo
clear
close data
return

```

Procedure PROCADD

```

clear
set talk off
set status off
store space(30) to mUsername
store space(2) to muserscode
use operator
do while .t.
  store "Y" to yn
  @ 2,27 SAY "Data Entry Form At Front Desk" color 'rg+/n'
  @ 4,6 TO 20,70 DOUBLE color 'rg'
  @ 6,10 SAY "OPERATOR'S CODE   : "
  @ 6,30 GET muserscode pict "XX"
  read
  locate all for muserscode = userscode
  if found()
    @ 8,10 say "Record Already Exist ...."
    wait ""
  else
    @ 8,10 SAY "OPERATOR'S NAME   : "
    @ 8,30 GET musername pict "@!"
    read
    @ 19,20 say "Are These Entries OK(Y/N) ?"
    @ 19,48 get yn pict "!"
    read
    if upper(yn) $ 'N'

```

```

        loop
    else
        if upper(yn)$"Y"
            append blank
            replace username with musername, userscode with
muserscode
        endif
    endif
    endif
    yn='Y'
    @ 19,20 clear to 19,60
    @ 19,20 say "Add more Records (Y/N) ?"
    @ 19,43 get yn pict "!"
    read
    if upper(yn) $ 'Y'
        clear
        store space(30) to mUsername
        store space(2) to muserscode
        loop
    else
        if upper(yn) $'N'
            exit
        endif
    endif
enddo
clear
return

```

Procedure PROCDEL

```

clear
set talk off
set status off
store space(30) to mUsername
store space(2) to muserscode
use operator
do while .t.
    store "Y" to yn
    @ 2,27 SAY "Delete records from File" color 'rg+/n'
    @ 4,6 TO 20,70 DOUBLE color 'rg'
    @ 6,10 SAY "OPERATOR'S CODE   :"
    @ 6,30 GET muserscode pict "XX"
    read
    locate all for muserscode = userscode
    if found()
        clear
        store "Y" to yn
        @ 8,10 SAY "OPERATOR'S NAME   :"

```

```

@ 8,30 GET musersname pict "@!"
@ 19,20 say "Delete this record(Y/N) ?"
@ 19,48 get yn pict "!"
read
if upper(yn) $ 'Y'
    delete
    pack
else
    exit
endif
else
    if .not. found()
        clear
        @ 7,10 Say "No Such Record Exist In Database, Press any key
to Continue..."
        set console off
        wait
        set console off
    endif
endif
enddo
clear
return

```

Procedure PROCMOD

```

clear
set talk off
set status off
store space(30) to mUsername
store space(2) to muserscode
use operator
do while .t.
    store "Y" to yn
    @ 2,27 SAY "Modification of Records" color 'rg+/n'
    @ 4,6 TO 20,70 DOUBLE color 'rg'
    @ 6,10 SAY "OPERATOR'S CODE   : "
    @ 6,30 GET muserscode pict "XX"
    read
    locate all for muserscode = userscode
    if .not. found()
        @ 8,10 say "Record Does not Exist ...."
        wait ""
    else
        store username to mUsername
        store userscode to muserscode
        @ 8,10 SAY "OPERATOR'S NAME   : "
        @ 8,30 GET musersname pict "@!"
    endif
endif

```



```

read
@ 19,20 say "Are These Entries OK(Y/N) ?"
@ 19,48 get yn pict "!"
read
if upper(yn) $'N'
  loop
else
  if upper(yn)$"Y"
    replace username with musersname, userscode with
muserscode
  endif
endif
endif
yn='Y'
@ 19,20 clear to 19,60
@ 19,20 say "Add more Records (Y/N) ?"
@ 19,43 get yn pict "!"
read
if upper(yn) $ 'Y'
  clear
  store space(30) to mUsersname
  store space(2) to muserscode
  loop
else
  if upper(yn) $'N'
    exit
  endif
endif
enddo
clear
close data
return

```

Procedure PROCVIEW

```

clear
set talk off
set status off
store space(30) to mUsersname
store space(2) to muserscode
use operator
do while .t.
  store "Y" to yn
  @ 2,35 SAY "Viewing Records" color 'rg+/n'
  @ 4,6 TO 20,70 DOUBLE color 'rg'
  @ 6,10 SAY "OPERATOR'S CODE   : "
  @ 6,30 GET muserscode pict "XX"

```

```

read
locate all for muserscode = userscode
if .not. found()
    @ 8,10 say "Record Does not Exist ...."
    wait ""
else
    store username to mUsername
    store userscode to muserscode
@ 8,10 SAY "OPERATOR'S NAME   : "
@ 8,30 GET musersname pict "@!"
read
wait ""
loop
endif
yn='Y'
@ 19,20 clear to 19,60
@ 19,20 say "View more Records (Y/N) ?"
@ 19,48 get yn pict "!"
read
if upper(yn) $ 'Y'
    clear
    store space(30) to mUsername
    store space(2) to muserscode
    loop
else
    if upper(yn) $ 'N'
        exit
    endif
endif
enddo
clear
close data
return

```

Procedure repl

```

clea
use operator
GO TOP
@01,(80-len(cbn1))/2 say cbn1
@02,(80-len(cbn2))/2 say cbn2
@4,18 say "REPORT GENERATION FOR LIST OF ACP OPERATORS"
@5,5 SAY REPLICATE("*",70)
@6,7 SAY "S/NO.          OPERATOR'S CODE          NAME"
@7,5 SAY REPLICATE("*",70)
SN = 0
DO WHILE .NOT. EOF()
    SN = SN + 1

```

```
        @ 7+SN,8 SAY STR(SN,4)+SPACE(14)+USERSCODE+SPACE(13)+USERSNAME
        SKIP
    ENDDO
? space(5),replicate("*",70)
wait " "
clear
close data
return
```

Procedure rep2

```
clea
use currency
GO TOP
@01,(80-len(cbn1))/2 say cbn1
@02,(80-len(cbn2))/2 say cbn2
@4,18 say "REPORT GENERATION ON TYPES OF CURRENCIES"
@5,5 SAY REPLICATE("*",70)
@6,7 SAY "S/NO.          CURRENCY CODE          CURRENCY NAME"
@7,5 SAY REPLICATE("*",70)
SN = 0
DO WHILE .NOT. EOF()
    SN = SN + 1
    @ 7+SN,8 SAY STR(SN,4)+SPACE(13)+C_TYPE+SPACE(13)+C_NAME
    SKIP
ENDDO
? space(5),replicate("*",70)
wait " "
clear
close data
return
```

Procedure rep3

```
clea
use DAILY
GO TOP
@01,(80-len(cbn1))/2 say cbn1
@02,(80-len(cbn2))/2 say cbn2
@4,18 say "REPORT ON TOTAL AMOUNT RECEIVED BY EACH OPERATOR"
@5,5 SAY REPLICATE("*",70)
@6,7 SAY "S/NO.    OPERATOR'S CODE    NAME
TOTAL-AMT"
@7,5 SAY REPLICATE("*",70)
SN = 0
TOT = 0
DO WHILE .NOT. EOF()
    SN = SN + 1
@ 7+SN,8 SAY STR(SN,4)+SPACE(7)+USERSCODE+SPACE(12);
```

```

+left (USERSNAME,25)+STR (AMOUNT,12,2)
TOT = TOT + AMOUNT
SKIP
ENDDO
@ 8+SN,5 SAY replicate("*",70)
@ 9+SN,35 SAY "TOTAL CASH DEPOSITED "+STR(TOT,14,2)
@ 10+SN,5 SAY replicate("*",70)
wait " "
clear
close data
return

Procedure rep4
clea
use DAILY
GO TOP
@01,(80-len(cbn1))/2 say cbn1
@02,(80-len(cbn2))/2 say cbn2
@4,10 say "REPORT ON TOTAL AMOUNT RECEIVED BASED ON FIT,UNFIT &
SUSPECT"
@5,5 SAY REPLICATE("*",70)
@6,7 SAY "S/NO.   OP-CODE           NAME           FIT           UNFIT
SUSPECT"
@7,5 SAY REPLICATE("*",70)
SN = 0
TOT1 = 0
TOT2 = 0
TOT3 = 0
DO WHILE .NOT. EOF()
    SN = SN + 1
    TOT1 = TOT1 + FITNOTES
    TOT2 = TOT2 + UNFITNOTES
    TOT3 = TOT3 + FAKENOTES
@ 7+SN,8 SAY
STR(SN,4)+SPACE(4)+USERSCODE+SPACE(5)+left (USERSNAME,20);
    +STR(FITNOTES,10,2)+STR(UNFITNOTES,10,2)+STR(FAKENOTES,10,2)
SKIP
ENDDO
@ 8+SN,5 SAY replicate("*",70)
@ 9+SN,10 SAY "TOTAL FIT, UNFIT & SUSPECT
"+space(6)+STR(TOT1,10,2);
    +STR(TOT2,10,2)+STR(TOT3,10,2)
@ 10+SN,5 SAY replicate("*",70)
wait " "
clear
close data
return

```

AUTOMATED CURRENCY PROCESSING (ACP)  
(A case Study of MICA Commercial Bank Plc. Maiduguri)

REPORT GENERATION FOR LIST OF ACP OPERATORS

\*\*\*\*\*

S/NO.	OPERATOR'S CODE	NAME
1	01	LABARAN KAGO
2	02	YUSUF MUHAMMED
3	03	ISAH BALEWA
4	04	HARUNA YOLA
5	05	CHARLES ADAMSON

\*\*\*\*\*

AUTOMATED CURRENCY PROCESSING (ACP)  
(A case Study of MICA Commercial Bank Plc. Maiduguri)

REPORT GENERATION ON TYPES OF CURRENCIES

```
*****  
S/NO.          CURRENCY CODE      CURRENCY NAME  
*****  
1              N5                FIVE NAIRA  
2              N10             TEN NAIRA  
3              N20             TWENTY NAIRA  
4              N50             FIFTY NAIRA  
*****
```

AUTOMATED CURRENCY PROCESSING (ACP)  
(A case Study of MICA Commercial Bank Plc. Maiduguri)

REPORT ON TOTAL AMOUNT RECEIVED BY EACH OPERATOR

```
*****
S/NO.   OPERATOR'S CODE   NAME                               TOTAL-AMT
*****
   1     01             LABARAN KAGO                       54860.00
   2     03             ISAH BALEWA                         24700.00
   3     05             CHARLES ADAMSON                     490400.00
*****
                                TOTAL CASH DEPOSITED                 569960.00
*****
```

AUTOMATED CURRENCY PROCESSING (ACP)  
(A case Study of MICA Commercial Bank Plc. Maiduguri)

REPORT ON TOTAL AMOUNT RECEIVED BASED ON FIT, UNFIT & SUSPECT

```
*****
S/NO.  OP-CODE  NAME                FIT      UNFIT     SUSPECT
*****
   1    01     LABARAN KAGO       54660.00  3600.00   200.00
   2    03     ISAH BALEWA       24500.00  4100.00   200.00
   3    05     CHARLES ADAMSON   490000.00 3000.00   400.00
*****
TOTAL FIT, UNFIT & SUSPECT      569160.00 10700.00   800.00
*****
```