The Implementation of Electronic Purse (E-Purse) System in the Banking Sector

(A case study of Afribank (Nigeria) PLC)

BY

AKINRINDE OLALERE

PGD/MCS/2004/2005/1175

DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE, FEDERAL UNIVERSITY OF TECHNOLOGY MINNA.

DECEMBER, 2006

The Implementation of Electronic Purse (E-Purse) System in the Banking Sector

(A case study of Afribank (Nigeria) PLC)

BY

AKINRINDE OLALERE

PGD/MCS/2004/2005/1175

A PROJECT SUBMITTED TO THE DEPARTMENT OF MATHEMATICS/COMPUTER SCIENCE FEDERAL UNIVERSITY OF TECHNOLOGY MINNA, IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE AWARD OF POST GRADUATE DIPLOMA IN COMPUTER SCIENCE.

DECEMBER, 2006

CERTIFICATION

This is to certify that this project, titled: "The Implementation of Electronic Purse (E-Purse) System in the Banking Sector has been read and confirmed by the undersigned as meeting the requirements of the Department of Mathematics/Computer Science, Federal University of Technology, Minna.

---------- ----------

PRINCE R.O. BADMUS DATE

PROJECT SUPERVISOR

---------- ----------

DR. I. A. AKINWANDE DATE

HEAD OF DEPARTMENT

---------- ----------

EXTERNAL EXAMINER DATE

DEDICATION

This project is dedicated to God for His affections and enablement, to my wife Mrs. Akinrinde and Children.

ACKNOWLEDGMENT

I wish to express my sincere gratitude to God who in His infinite mercy made it possible for me to complete this programme.

Special thanks to my Supervisor, Prince Badmus R. O. who, despite his busy schedule, could afford to read the manuscripts, gave guidance in the course of writing this project and subsequently gave his approval. I also wish to thank my Head of Department Dr. I. A. Akinwande, my course coordinator Alhaji Ndanusa A. and all my Lecturers in the department for their immense contributions towards the success of this program.

I also appreciate the encouragement of my wife Mrs Akinrinde whose support has been so enormous. Appreciations to my children for their endurance throughout the period that the program lasted.

My sincere gratitude goes to Engineer Akeem Abiona for taking pains to read through the papers and made appreciable contributions.

I wish to thank all my course mates, brothers, sisters, and friends who contributed in various ways towards the successful completion of this work.

ABSTRACT

Electronic Purse (E-Purse) also known as **Smart-cards** act as an electronic payment alternative to bank notes. Automated currency solutions offer greater levels of convenience to consumers, incremental sales opportunities and reduce operating costs to merchants and new service revenue streams to banks.

Smart card applications reduce operating costs and control fraud in electronic benefits programs, carrying patient information for healthcare applications and providing a secure vehicle for delivering government benefits such as social insurance and welfare programs.

Smart cards (e-purse) are used in a variety of other applications that require greater processing power and/or more secure storage. College students use smart cards (e-purse) to pay for cafeteria and bookstore purchases and to access health, recreation and other services; commuters use smart cards (e-purse) to pay tolls and parking fees; and parents use smart cards (e-purse) to pay for child care.

TABLE OF CONTENTS

Chapter 1

INTRODUCTION

Chapter 2

LITERATURE REVIEW

Chapter 3

NEW SYSTEM DESIGN

Chapter 4

SYSTEM IMPLEMENTATION

Chapter 5

Chapter 1
INTRODUCTION

1.1 The Electronic Purse (E-Purse) Introduction

An Electronic purse is a Smartcard that holds an electronic equivalent of cash. The electronic purse can be used for the payment of goods and services i.e. the electronic purse is a substitute for bank notes.

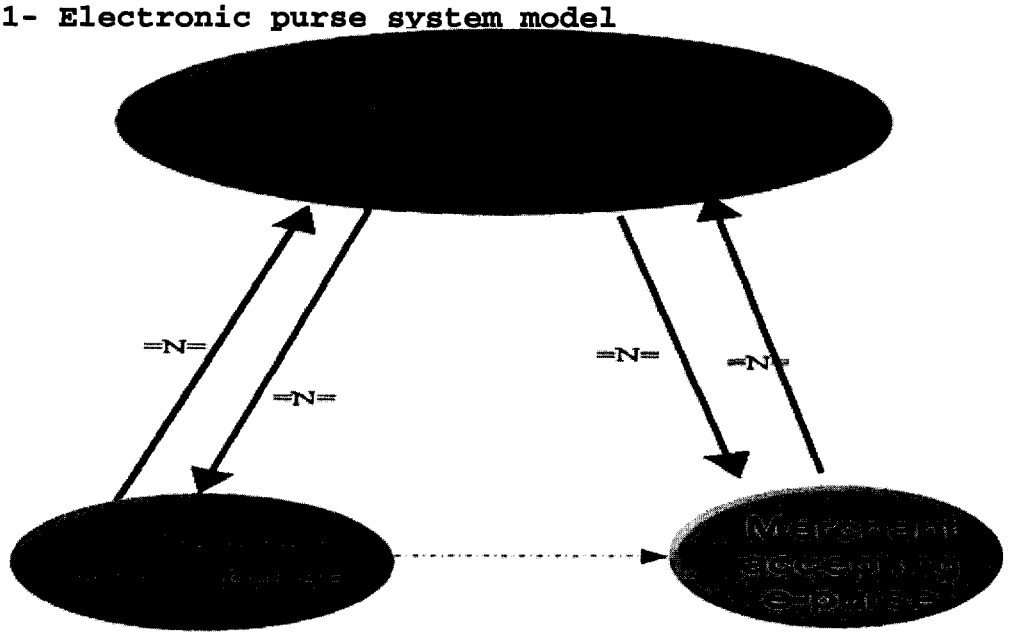
Interest is fast developing in a multipurpose prepaid smart card commonly known as the **electronic purse**. In contrast to a debit card, the electronic purse is intended to facilitate a variety of small-value retail transactions and so it is a clear substitute for currency. It might function as follows.

Monetary value would be loaded onto the card, with a corresponding debit to the cardholder's account at a financial institution. In a retail transaction, monetary value would be transferred from the purchaser's card into the merchant's terminal in an off-line mode. The value of consumer purchases made with electronic purses would accumulate in the merchant's terminal and would be transferred to the merchant's account at a financial

institution from time to time through on-line transactions.

More technically, in many purse systems the financial institution issuing the card would earmark or put a hold on an amount of funds in the cardholder's account that is equivalent to that recorded on the smart card, and would in effect be providing a guarantee of the value shown on the card. When a transaction is cleared through the payments system, a debit would be made to the cardholder's special suspense account, and a credit would be made to the merchant's bank account. This type of electronic purse would essentially function as an off-line debit card (Figure 1).

Figure 1- Electronic purse system model



In most proposed systems, the cardholder would be able to "replenish" the monetary value on the card at Issuer's bank. Such electronic purses would have to be equipped with personal identification in order to keep track of individual transactions. This feature would also assure the cardholder of enhanced security.

Other purse systems are being designed that share physical currency's characteristic of anonymity; they would be an even closer substitute for cash. Any institution issuing this type of electronic purse would have to establish a general suspense account for the amount outstanding in its issued purses. In such systems, monetary value could be transferred directly between cards without the action of an intermediary.

1.2 Definition Of Terms

Smartcard is a new concept in the Nigeria Payment and Settlement System. It is therefore, necessary to start this overview with definition of some of terms used.

The following terms are in common use:

- Smartcard
- Electronic Purse

- Cardholder
- Issuer Bank
- Acquirer Bank
- Merchant
- Merchant/Transport Card
- Bank Teller Terminal (BTT)
- Point of Sale Terminal (POS)
- Hotlist

1.2.1 SmartCard

The term 'Smartcard' applies to all a technology that has become a generic term for all sorts of advanced card technology. A card is in accordance with International Organization for Standardization (ISO), a piece of nominal dimension:

85.6mm x 53.9mm x 0.76 mm

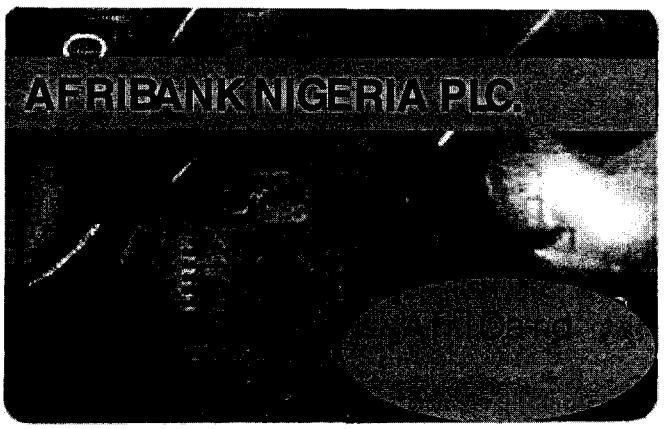
The term 'Smart' applies to a card that carries a semi-conductor chip. The chip contains basic features of a computer and thus has a small processor, storage facility, an operating system and set(s) of stored instructions called programs. Thus, Smartcard is a card of ISO dimensions which has in-built logical ability. Smartcard range from small capacity memory only chips

(mainly used for Phone cards) to large (16k bytes) memory capacity microprocessor chips and more recently, microprocessor chip with an added high speed arithmetic unit for use on public key cryptographic calculations such as Random Security Algorithm(RSA) and Data Encryption Standards.

There are basically two types of Smartcards viz: Disposable and Re-loadable cards. The **disposable card** is that is disposed off after user, example is Phone cards, while **re-loadable card** allows users to replenish the value of their card as often as they wish.

1.2.2 Electronic Purse

An electronic purse is a device (smartcard) that holds an electronic equivalent of cash. The electronic cash is commonly accepted for payment for goods and services at designated Point of Sale (POS) outlets. Values are loaded onto the card in a bank and unloaded (deducted/debited) at the Point of Sale outlets in payment for goods and services. The purse can only be used while its balance is positive.



- **Smartcard (Electronic Purse)**

1.2.3 Cardholder

A cardholder is the bank customer who having purchased and loaded a card in a bank will use the card in payment for goods and services.

1.2.4 Issuer Bank

Issuer bank is the bank that issues the card to its customers. It obtains the card from the Smartcard company, embosses them to identify users and thereafter, issues them to the customer.

1.2.5 Acquirer Bank

This is the bank that negotiates and enters into agreement with merchants (shop owners) to accept

Smartcard(Electronic Purse) in payment for goods and services.

1.2.6 Merchant

A merchant is a service provider/trader, who has Point of Sale (POS) terminals installed in their shops, offices or outlets for Smartcard (electronic purse) transactions.

1.2.7 Merchant Card or Transport Card

A merchant card or transport card is issued to the merchant by an acquiring bank. The card is issued to sum up periodic transactions. All transactions are uploaded from the merchant POS terminal to a Bank Teller Terminal (BTT) via a merchant card or modem if on-line. The merchant card is also used to transfer the hotlist to the merchant POS terminal.

1.2.8 Bank Teller Terminal (BTT)

The Bank Teller Terminal is used to interact with cardholders and merchant cards. It will be located in the bank branches and will be operated by bank employees, who will provide services to cardholders and merchants. The Bank Teller Terminal will have an on-line connection to the Server.

The functions provided by the BTT are as follows:

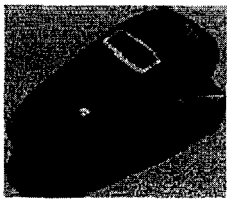
- Exchange of value on card for cash.
- Loading value onto card.
- Transaction amount entry.
- Off-line PIN validation.
- Card identification and validation.
- Card balance checking and inquiry
- Print or view customer card transaction history.
- Key maintenance.
- Destroy hotlisted cards inserted in the terminal.
- Receipt printing.
- Transaction reports, summary and details.
- Upload of transactions from a merchant card.
- Download hotlist to merchant card.
- Issue card to cardholder, i.e. cardholder selects own PIN.

- **Bank Teller Terminal (BTT)**

1.2.9 Merchant Point of Sale Terminal

The merchant Point of Sale terminal is used to store payment transactions once value has been removed from a cardholder's card as payment for goods and services. Each Point of Sale device within the system will have a unique identification number. The Point of Sale terminal will support the following on-line and off-line functions:

- Exchange of value for goods and services
- Exchange of value on card for cash
- Receipt printing
- Transaction amount entry
- Off-line PIN Validation
- Destroy hotlisted card inserted on the machine
- Card identification and validation
- Upload hotlist from merchant card
- Card balance checking and inquiry
- Printing customers' card transaction history
- Key maintenance
- Transaction reports, summary and details
- Upload transaction files via modem
- Download of hotlist via modem.



- **Point of Sales (POS)**

1.2.10 Hotlist

Hotlist is a list of lost or stolen cards made available to bank's branches connected to the hub. When a card is reported lost or stolen, the card is hotlisted. The uploaded hotlist is then transferred to the merchant terminals through the merchant card.

1.3 History Of Afribank Nigeria Plc.

AFRIBANK received banking license on October 20, 1959, and opened the first branch in the city of Kano on January 4, 1960. It was then known as Banque International Pour L'Afrique Occidentale (BIAO).

The Bank's Headquarters was opened in 1960 at 94 Broad Street, Lagos. The head office was transferred to 51/55 Broad Street in 1998 following the completion of the Afribank Plaza. The Bank opened two more branches in Port Harcourt and Aba just before the outbreak of the Civil War in 1967. On May 30, 1969, International Bank for West Africa (IBWA) was incorporated in the country as a limited liability company.

On account of Nigeria Enterprises Promotions Decree, the Federal Military Government in 1976 acquired 60 per cent of the company's equity structure, while the remaining 40 percent was acquired by all foreign banks operating in Nigeria as at that time. By December 1976, the Bank got her share capital licensed of ₦3.6million. The year 1979 saw the paid up capital of IBWA/AFRIBANK increase further to ₦15million. Also, that year, for the first time, a Nigerian, Mr. Oladele Olashore was appointed the Bank's Managing Director. In 1980, the Federal Government gave up 10 percent of her equity holding in the Bank to staff. The Bank changed its name finally to Afribank Nigeria Limited with effect from January 1, 1990.

AFRIBANK Nigeria Plc is one of the four biggest banks in Nigeria. From a humble beginning of one branch in 1960,

the bank has grown to be represented in 137 locations in major and sub-urban settlements throughout the country. The Afribank group stands out as the most diversified in the banking sector. Besides the commercial bank, which is the parent company, the Afribank group also owns Afribank International Limited, a full service merchant bank, ANP International Finance Company, an offshore finance company, AIL Securities Limited, a stock broking firm, Afribank Insurance Brokers, an insurance brokerage firm, Afribank Trustees and Investment Limited, a trustee and Investment Company and Afribank Estate Company Limited, an estate development company.

To complement its fully diversified financial business empire, Afribank also made substantial investment in Consolidated Discount House Limited, Central Insurance Company (through Afribank Insurance Brokers), First Securities Discount House Limited, Afrexibank, Cairo, Electric Meter Company of Nigeria and Smartcard Nigeria Limited.

The development of this financial superstructure has made it possible for the bank to leverage as a one-stop shop for a wide range of financial services.

1.4 Problem Definition

The following problems are being encountered with the present Nigerian payment system.

- Some of the present payment system involves too much cash carrying for exchange of goods and services.
- Lack of trust on some of the payment instrument.
- Insecure payment instrument.
- Faking, forging and cloning of the payment instruments.
- Movement of large sum of cash exposes the carriers to imminent danger in the hand of hoodlums.

1.5 Objectives

The objectives of this project work are:

- To provide alternative to the present payment system that eliminates too much cash carrying for exchange of goods and services.
- To provide a secured payment instrument.
- To make payment for goods and services easier.
- To provide a reliable and trusted instrument.

1.6 Limitation

- This study is concerned with the mode of payment through electronic purse (smartcard) within Afribank Nigeria Plc only.
- The payment currency is limited to only Naira and is within Nigeria.
- Of all Smartcard payment system, only electronic purse is considered.
- Due to the cost of acquiring the Bank Teller Terminal (BTT) and Point of Sales machine (POS), Afribank model were used for the test running of the software.

1.7 Methodology

This project work is done by studying and reviewing the present payment system in Nigeria, other products that are related to the electronic payment system.

Chapter 2

LITERATURE REVIEW

2.1 The Nigerian Payment System

A country's payment system comprises of all items used in payment for goods and services in the country. It is dynamic and changes over time, depending on the level of economic activities, the sophistication of the financial and banking system and level of financial literacy.

As financial transactions in Nigeria are predominantly cash-based, with attendant risks and high handling costs. The need for a more secured and convenient means of payments has led, many banks in recent years to introduce alternatives. Presently, the Nigerian payment system comprises:

- Money (Coins & Bank Notes)
- Personal Cheques
- Certified Cheques
- Bank Drafts
- Bankers Payment
- Mail Transfers
- Electronic Funds Transfer
- Automated Teller Machines

- Plastic Money/Card (Credit Cards, Smartcard)

The most recent of these payment systems is the Smartcard technology used in certain parts of the world today, with Nigeria not left out this time around. Smartcards are already gaining grounds in Nigeria through Valucard, Smartpaycard, ESCA, and Paycard.

2.2 Evolution of Smart Cards

The smart card technology originated in both France and Japan and much of the impetus for its development has come from the national governments of these two countries. In France, the Director General of Telecommunications, faced with modernizing the national telephone system in 1974, decided that using smart cards would be a good way to update its pay phone system. It was also felt that this new technology could be a key factor in responding effectively to an expected strong growth in demand for home banking and shopping services. Furthermore, the French banks were interested in smart cards because of an earlier explosive increase in the issuance of cheques and because of a problem with fraud related to their Automated Teller Machines which

operated, generally speaking, in an off-line mode. Finally, the French government had invested substantially in computer research and development in the second half of the 1970s, contributing still further to the development of the smart card. In Japan, the national government placed special emphasis on the role of computer technologies in its national economic development programs. The subsequent emergence of a large computer-chip manufacturing industry in Japan also contributed to the development of smart card applications in that country.

Smart cards have found extensive applications in United State of America; with an estimated half a billion cards in use worldwide as of 1994 (Ravensbergen 1994). The diffusion of smart card payments applications has been particularly rapid in Europe and East Asia, reflecting in part the relatively higher cost of telecommunications services in those countries than in North America. At the moment, most prepaid cards are employed for single purpose transactions, although interest in the electronic purse application is growing rapidly. Many electronic purse pilot projects have been announced over the past year, both in North America and overseas.

Still more recently, there has been interest in developing payments systems for use on the Internet and other personal computer networks (Crone 1995; Holland and Cortese 1995). While some of these systems simply involve the use of bank and credit card numbers for purchases on the Internet, other systems are to include the "virtual" equivalent of an electronic purse card.

2.3 Operation Of Africard Product

2.3.1 Africard Product

The Smartcard scheme will offer a re-loadable, EMV complaint, electronic purse, branded AfriCard, aimed at cash substitution, by providing an electronic means of payment for goods and services at the Point of Sale.

The electronic purse will be PIN protected and receipt will be produced after each credit and debit of the purse. The last ten transactions will be stored on the card.

2.3.2 System Components

A number of separate components operate together to create the complete Africard E-purse Payment System. The components include:

- Cards
- Merchant POS terminal
- Banks Teller Terminals
- Africard Management System

2.3.3 AfriCard Security

- **Personal Identification Number (PIN)**

PIN is a set of codes (numbers) selected by any cardholder as security lock/key for access to his card either at points of loading or payment. If it is forgotten, only the issuing bank can undo the codes and give the cardholder another opportunity to select new set of codes as security to his card.

- **Personalization**

Cards will be electronically and physically personalized by issuing bank (Afribank) at the bank's head office. Electronic personalization will involve printing the card number, cardholder name and expiry date on front of the card.

- **Information Privacy**

Information stored in any card cannot be made available to anyone except with the knowledge/consent of the owner. This is the essence of PIN, which serves as the gateway to access the card for whatever data manipulation.

- **Hotlist**

This is a document containing series of cards reported lost or stolen. This list instructs POS terminals to disregard information on such cards and to destroy such cards. The hotlist is distributed to all merchants, who in turn load the refreshed list into the POS.

- **Risks**

Risks to all participating partners have been identified, isolated and minimized to the least conceivable.

2.3.4 Africard Operations

The operations of the Africard Management System include the following:

- Cardholder/Merchant Registration
- Card Issue

- Card Balance
- Cash Load/Unload
- Change PIN/Unlock PIN
- Upload Transaction from BTT
- Upload Transaction from Merchant/Transport Card
- Hotlist Management
- Reports
- Users Configuration
- Bank's Liability Management

Cardholder/Merchant Registration

This module handles the registration of both New Cardholder and Merchant. Information on the Cardholder/Merchant form are extracted and input through this module.

Card Issue

Card will be issued through the cardholder's branch on application by him/her. The branch will forward the application, including customer information, the branch code and account to which the card is linked to the bank's head office. On personalization, the cards are then delivered to the branch. The customers will collect

the cards at their branches where they will be required to enter their own PIN, which replace the default PIN created on the cards when they were personalized.

Card Balance

It is possible for cardholder to check his balance through the branch Bank Teller Terminal, merchant Point of Sale and card reader device.

Cash Load

Cardholders may be able to load at all branches of their bank if the branches are on-line. Once a card has been issued, the cardholder may load value onto his/her card, using the Bank Teller Terminal in the bank branch. The cardholder will insert the card into the PIN pad attached to the Bank Teller Terminal; the customer service officer will enter the amount to be loaded. The bank's liability is adjusted less the cash loaded amount. When the transaction is completed, the value is loaded onto the card and a receipt is printed.

Cash Unload

Cardholder may decide to get cash from his card at the bank's branch. The card is inserted, the officer enter

the amount to unload, after verifying, the cardholder will enter the Personal Identification Number (PIN). The system check whether the PIN entered is correct, if yes, the value is unloaded and a receipt is printed. The bank's liability is adjusted with the cash unload amount.

Change PIN/Unlock PIN

It is possible for cardholder to change his Personal Identification Number (PIN) at any time provided he knows the old PIN. If he lost his PIN there is an option called Unlock PIN that is used to introduce new PIN without asking for old PIN.

Upload Transactions from BTT/Transport Card

All transactions of Bank Teller Terminal and Point of Sale are uploaded to the central database through the Upload transactions option. This update the master files of all records and tables affected.

Hotlist Management

Issuing banks will place cards reported lost or stolen on a hotlist. Placing card on hotlist ensures that the card will be destroyed if used. Branch Bank Teller Terminal download hotlist online at least once in a week while the

Merchants' Point of Sale received Hotlisted Cards through the Transport card.

Reports

The following reports are possible with the Africard Management System.

- Daily Transaction Report
- Cardholder statement report
- Hotlisted Cards
- Active Cards
- Expired Cards

Administration

New users are first introduced to the system through the Users configuration modules. Menu are also assign to new users. Modification and deletion is done here.
Liability Management is also under this module.

2.4 Benefits Of Africard

2.4.1 Benefits for Bank

- Improved payment system through replacement of cash
- Reduced cash handling costs
- Improved transaction times at branches

- Reduced security risk for staff
- Improved customer service image.

2.4.2 Benefits for Merchants

- Safer and more secure payment system to customers
- A better way of documenting transactions as each of them is receipted automatically and added up at end of each business period.
- Reduction in the volume of cash handled at the shops.
- Outlet managers will have better accountability and cash pilferage will be eliminated.
- Offers opportunity to design special loyalty schemes for customers.
- Increase in customer base.
- Retention of market share and improved market positioning
- Faster transaction time
- Cost saving from reduced infrastructure equipment (Note counting machine, Note binder, Counterfeit detector, etc.)
- Enhanced corporate image
- Match competitor's offering.

2.4.3 Benefits for Customers

- Increase security
- Reduced handling of cash
- Greater convenience

2.5 Barriers To The Development Of E-Purse Payment Systems

In spite of the strong interest worldwide in the electronic purse, the adoption of this innovation has been comparatively slow. There are a number of reasons for this.

- First, the market for the electronic purse, like other innovations that involve the creation of networks between the suppliers of services and their customers, needs to attain a critical mass before the purse can be used effectively. Prior to this stage, there will be considerable uncertainty among both consumers and merchants as to the potential usefulness of the product. Clearly, the benefits to consumers will rise as the new means of payment becomes acceptable to

merchants, while the benefits to merchants will rise with greater usage by consumers.

- Second, lack of good telecommunications network affects the connectivity of the smart cards networks.

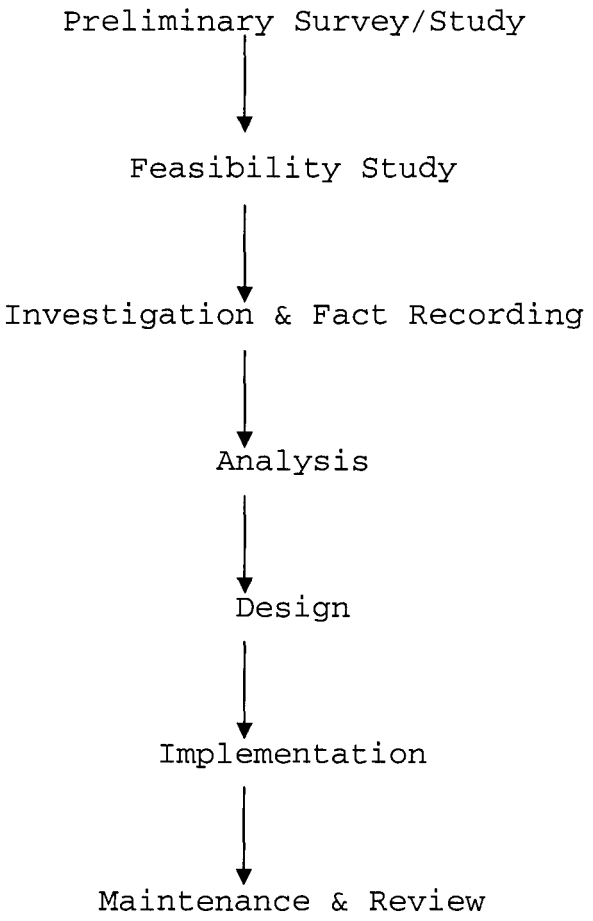
- The average consumer may also prove to be slow to accept the electronic purse, because cash is still such a basic feature of daily life. Many people may consider an electronic purse more complicated to use than cash and may have concerns about the risk of card or equipment failure or about difficulties in recharging an electronic purse. Others may also be bothered by the potential loss of privacy, particularly in the case of multifunction applications, where a variety of personal information might be stored on a single card.

- Lack of experience with Electronic payment systems
- Lack of suitable merchants
- Ignorance regarding security feature
- Cost of training.

Chapter 3
NEW SYSTEM DESIGN

3.1 Methodology

The methodology adopted for the proposed system is the System Life Cycle approach as shown below:



3.1.1 Preliminary Survey/Study

The purpose of this survey is to establish whether there is a need for a new payment system and how cash carrying Nigeria society can be turned to cashless society.

3.1.2 Feasibility Study

The purpose of the feasibility study is to investigate the alternative to the present Nigerian payment system i.e. electronic purse payment system and how it is being operated in other countries, the cost and benefits analysis.

3.1.3 Investigation & Fact Recording

Detailed study on the present and proposed system is conducted. This is a more detailed and comprehensive than feasibility study.

3.1.4 System Analysis

Analysis of the full description of the present payment system and of the objectives of the proposed system is done here.

3.1.5 System Design

Next to the System Analysis is the Design phase i.e. the designing of the proposed system called Electronic Purse/Smartcard Payment System. A system specification is produced.

3.1.6 Implementation

This system is implemented using a modern day development tools and a robust database engine. Microsoft Visual Basic is used for the front-end, SQL Server 2000 for the backend and Crystal Report for the reporting tool.

3.1.7 Maintenance & Review

After the system is implemented and operational, it is examined to see if it has met the objectives set out in the specifications.

3.2 Input Design

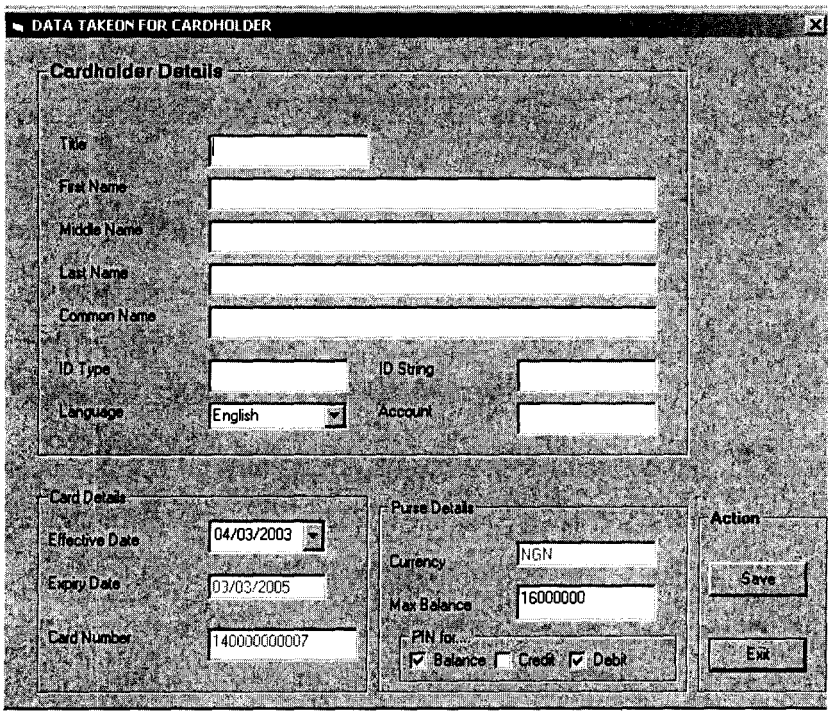
This is the means where data is transferred into the central processor. The adopted method is by using keyboard to key data directly to the system.

Basically there are 4 input Forms.

- Cardholder Details Form
- Branch/Merchant Form
- Cash Load Form
- Cash Unload Form

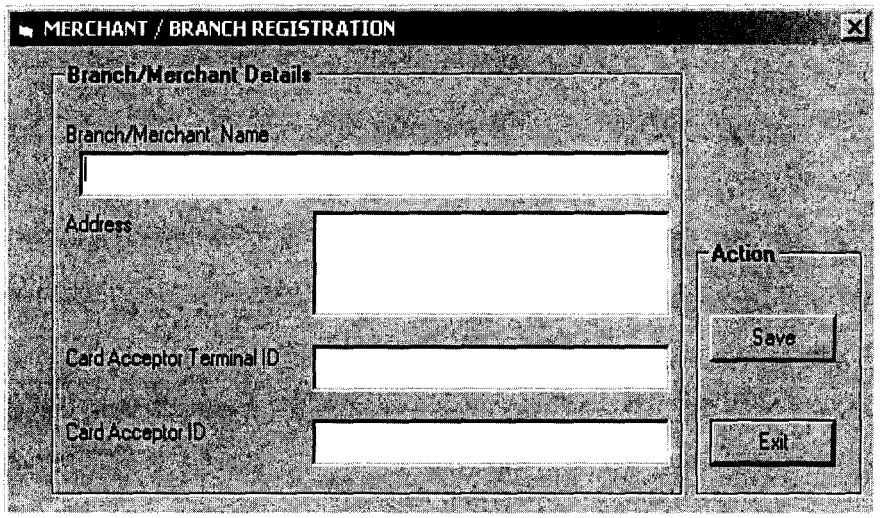
i) Cardholder Details Form

Purpose: This is used to capture data of new cardholder.



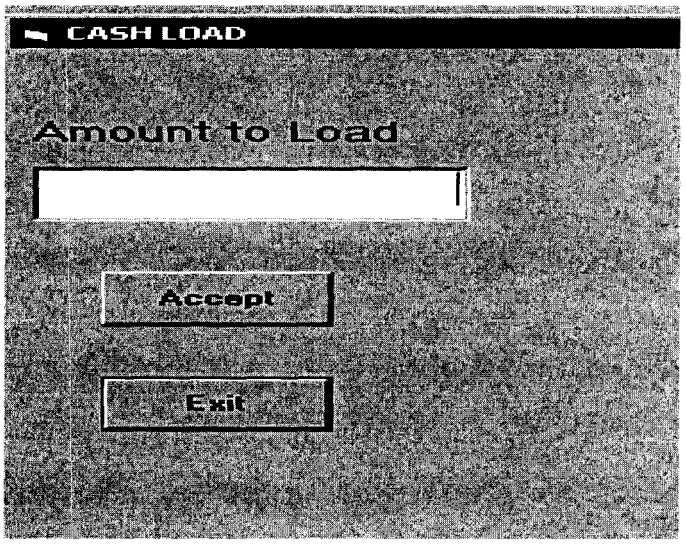
ii) Branch/Merchant Form

Purpose: For the registration of new branch or merchant.



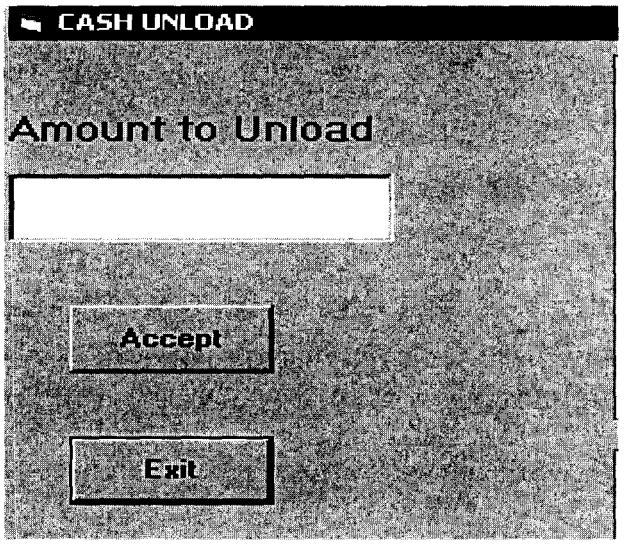
iii) Cash Load Form

Purpose: Use for loading of cash to the card.



iv) Cash UnLoad Form

Purpose: Use for unloading of cash to the card.



3.3 Output Design

Output is the means by which computer communicate result to the users for decision making. This can be through Monitor(soft copy) or Printer (hard copy).

Reports available:

- Daily Transaction Report
- Card Holder Statement
- Hotlisted Cards
- Active Cards
- Expired Cards

i) Daily Transaction Report

This is a daily transaction report. It gives details of all transactions (Cash Load and Cash unload) for a day. It contains the following columns:

- Transaction ID
- Card Number
- Card Acceptor Terminal ID
- Card Acceptor ID
- Amount
- Transaction Type
- Date/Time of transaction

ii) Card Holder Statement

This report shows all the transactions the cardholder has done. It consists of the following:

- Date/Time of transaction
- Transaction type
- Branch/Merchant Name
- Debit
- Credit

iii) Hotlisted Cards report

This shows the list of all hotlisted cards. It contains the following columns:

- Card Number
- Card Holder Name
- Hotlisted Date

iv) Active Cards

This is list of all active cardholders.

- Card Number
- Cardholder Name
- Expiry Date

v) Expired Cards

This is the list of all expired cardholders.

- Card Number
- Cardholder Name
- Expiry Date

3.4 File Design

This describe how the data is to be structured and physically stored on backing storage device.

DATABASE : EPURSE

FILENAME: EPURSE_DATA.MDF

DATABASE ENGINE: SQL SERVER 2000

STORAGE MEDIA: HARD DISK

- TABLE NAME: **CardHolderDetails**

  PURPOSE: This holds the details of the Cardholders.

  PRIMARY KEY : ***CardNumber***

Column Name	**Data type**	**Length**	**Description**
Title	Varchar	10	Title
FirstName	Varchar	20	First Name
MiddleName	Varchar	20	Middle Name
LastName	Varchar	20	Last Name
IDType	Varchar	15	Identification Type
IDString	Varchar	15	Identification String
Language	Varchar	10	Language
Account	Varchar	13	Account Number
EffectiveDate	Datetime	8	Effective Date
ExpiryDate	Datetime	8	Expiry Date
CardNumber	Varchar	13	Card Number
NCurrency	Varchar	16	Nigerian Currency
MaxBalance	Numeric	9	Maximum Balance
PinBalance	Varchar	1	Pin Balance
PinCredit	Varchar	1	Pin Credit
PinDebit	Varchar	1	Pin Debit
Hotlist	Varchar	1	Hotlist
HotlistedDate	Datetime	8	Hotlisted Date

- TABLE NAME: **TransactionDaily & TransactionMaster**

  PURPOSE: TransactionDaily keeps transactions pending the time it will be uploaded while TransactionMaster contains all transactions.

  PRIMARY KEY : ***TransactionID***

Column Name	**Data type**	**Length**	**Description**
TransactionID	Varchar	20	Transaction Identification
TransactionType	Varchar	11	Transaction Type
CardNumber	Varchar	13	Card Number
Catid	Varchar	15	Card Acceptor Terminal Identification
Caid	Varchar	15	Card Acceptor Identification
DateTime	Datetime	8	Date/Time of Transaction
Amount	Numeric	9	Amoount
UserId	varchar	12	User Identification

- TABLE NAME: **BranchMerchant**

  PURPOSE: Keeps the records of Branches and Merchants.

  PRIMARY KEY : ***Catid***

Column Name	**Data type**	**Length**	**Description**
BranchMerchantName	Varchar	60	Branch/Merchant Name
Address	Varchar	200	Address
Catid	Varchar	15	Card Acceptor Terminal Identification
Caid	Varchar	15	Card Acceptor Identification

- TABLE NAME: **BranchLiability**

  PURPOSE: Bank Liability entry is kept here.

Column Name	**Data type**	**Length**	**Description**
LiabilityAmount	Numeric	9	Liability Amount
AmendmentDate	datetime	8	Amendment Date

- TABLE NAME: **User_tab**

  PURPOSE: Keeps the records of users.

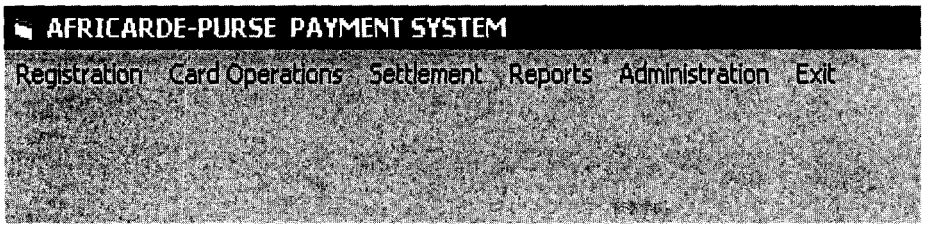
  PRIMARY KEY : ***Userid***

Column Name	**Data type**	**Length**	**Description**
User_name	Varchar	64	User Name
User_id	Varchar	12	User Identification
Staff_id	Varchar	5	Staff Identification No.
Start_date_profile	Datetime	8	Profile Start Date
End_date_profile	Datetime	8	Profile End Date
User_class	Varchar	20	User Class
User_branch	Varchar	30	User Branch
User_pass	Varchar	12	User Password
Remarks	Varchar	16	Remarks

3.5 System Specification

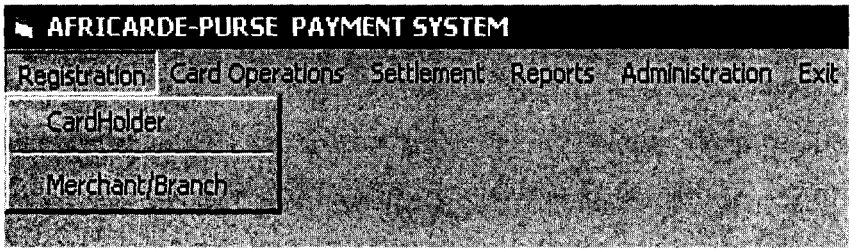
The system specification describes the new system. The software for the new is known as **AfriCard E-Purse Payment System**. This system is a windows based system using a menu and submenu technique.

- **MAIN MENU**



- **REGISTRATION MENU**

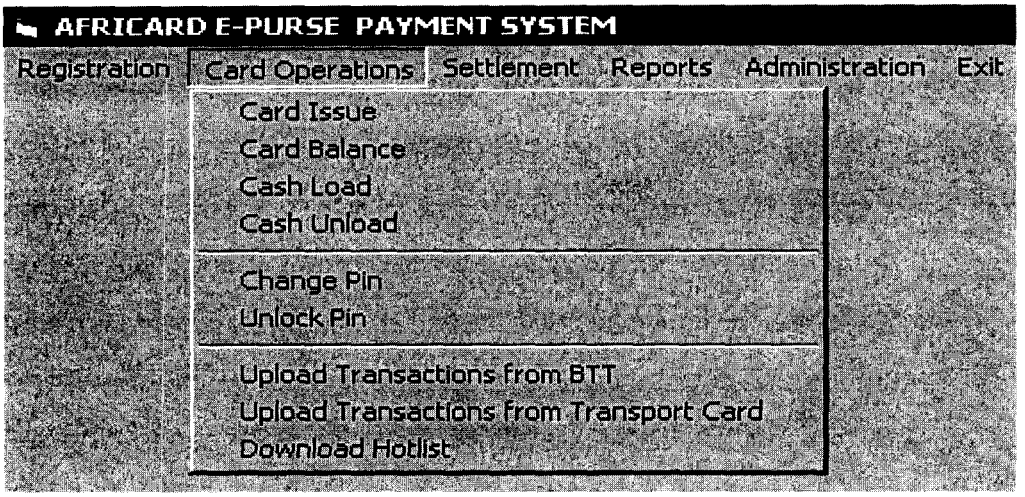
This registration menu is used for the capturing of Cardholder and Branch/Merchant Data into the system.



- **CARD OPERATIONS**

This menu handles the following operations:

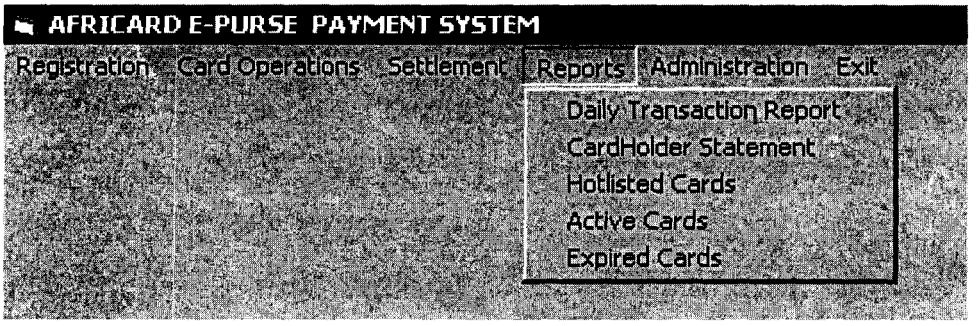
  - Card Issue
  - Card Balance
  - Card Load
  - Cash Unload
  - Change PIN
  - Unlock PIN
  - Upload Transactions from BTT
  - Upload Transactions from Transport Card
  - Download Hotlist



- **REPORTS**

This produce reports for the following:

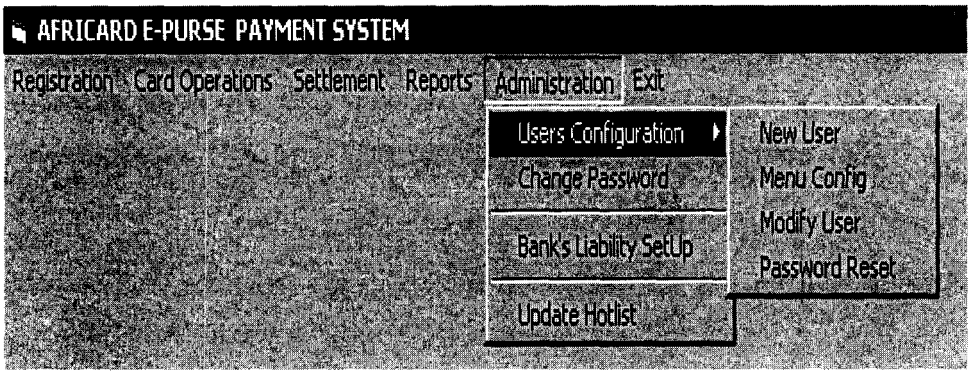
- Daily Transaction Report
- CardHolder Statement
- Hotlisted Cards
- Active Cards
- Expired Cards



- **ADMINISTRATION**

This is used for setup of the following:

- o Users Configuration
- o Change Password
- o Bank's Liability Setup
- o Update Hotlist



3.6 System Security

Security is the protection of data against unauthorized access against accidental or intentional or destruction of data. While designing the system, the following essentials were put into mind:

- o Users must be created by the System Administrator.
- o Users must be positively identified by the system before they are given access to the data.

3.7 COSTS AND BENEFITS ANALYSIS

COSTS

The overall cost of producing smartcards depends on how long the cards last and how many cards are needed.

The project of smart cards production is comparable to any other stationary or security printing, in that, the higher the number of copies required, the cheaper the cost. This is so, because both the hardware and software that are installed to produce a single smart card is what is needed to produce a million copies. The only additional cost involved is the length of period of production, quantity of card and energy dissipation.

However, below is the approximate figurative analysis of the production of one single smart card. It is assumed here that, smart card facilities are just being introduced to a particular populace.

(i) Cost of carrying out feasibility study - ₦2.5m

(ii) Cost of carrying out public enlightenment - ₦1.5m

(iii) Cost of preparation of software - ₦0.5m

(iv) Cost of procurement of hardware - ₦2.0m

(v) Cost of other accessory equipment - ₦1.5m

(vi) Cost of rent/erection of stations - ₦15m

(vii)	Cost of hiring personnel -	₦10m
(viii)	Cost of procuring security cards -	₦5m
(ix)	Cost of providing training for personnel-	₦3m
(x)	Logistics -	₦2m
(xi)	Miscellaneous expenses -	₦5m
	Total	₦48m

Above is an approximate figure for the production of a single unit of smartcard. However, to produce 100 million smartcards may not exceed ₦55m.

From the analysis above, it will be observed that smartcard project is capital intensive. Meanwhile, the benefits accrued are enormous.

BENEFITS

Deploying Smart Card login to your network will provide the following benefits:

- It is an alternative to banknotes
- The Automated Currency solutions offers greater level of convenience to customers

- It provides increased sales opportunity and reduce operating cost of merchants and control fraud
- It carries holders health information for health care applications and provides secure vehicle for delivering government benefits such as social insurance and welfare programs
- It reduces the risks of carrying life banknotes around in exchange of services
- It protects the privacy of the cardholders
- It improves business transaction time.
- The card store information, money and or application that can be used for banking/payment, loyalty and promotion, access control, ticketing, store value, identification, parking and toll collection.
- Smart cards provide powerful authentication to prevent misuse of your resources.

Chapter 4

4.0 SYSTEM IMPLEMENTATION

4.1 Choice of Language

The choice of programming language used in this project work is Visual Basic 6.0

This is because Visual Basic is able to compile on any system that runs windows of any version. Since the work comprises the use of database management system and the data so stored, has the ability to grow by the day.

Visual Basic is most suitable as it can handle large independent data conveniently. Independent, in that, it only acts as user interface, not having direct contact with the raw data, rather it exposes object that manipulate this data regardless of the database management system used.

Another driving factor for using Visual Basic for this works its portability. The language has the ability to execute on all windows running machine as it builds it executables and library files based on windows architecture. One does not need to configure windows to accept program built with Visual Basic 6.0

4.2 Features of Language chosen

Visual Basic 6.0 is one of the 32bits and most efficient windows application development languages. It provide the developer with ease of using Graphic Users Interface that enables programmers develop complex window forms by dragging and dropping controls on the forms.

Visual Basic supports multi threading and communicates easily with windows.

It supports optional code compilation to either favour the processor or memory.

With the invent of active x Visual Basic can now handle classes and other complex data structures.

Other applications can be run with the Visual Basic environment by either late or early binding mechanism of Visual Basic.

In Visual Basic object names are as long as 256 characters, this makes it easy and give meanings to object names.

4.3 Changeover/Conversion Procedure

This is simply moving from old system to a new system. There are various changeover methods, such as :- Direct changeover, parallel changeover and pilot / piecemeal changeover

Direct Changeover is discarding the old system completely and immediately using the new system, thus the old system is discontinued altogether and new system becomes operational immediately.

Parallel Changeover is using both old and new system until it is confirmed that, the new system performs correctly, thus the old and new systems are run concurrently using the same inputs. The outputs are compared and reasons for differences resolved. Outputs from old system continue to be distributed until the new system has proved satisfactorily. At this point the old system is discontinued and the new one takes its place.

Pilot Changeover is having only a small group of people over a period of time to establish its reliability and acceptance. Upon confirmation, the remaining section of the populace is introduced into the new system. In this method the test period can either be run in parallel with the existing system or in direct changeover. This method is the same as the piecemeal method. It has an advantage of showing earlier, all potential implications in the new system to enable the proponent device appropriate solutions.

4.4 Software & Hardware Requirements

For successfull implementation of Africard E-purse payment system, the following equipments are required:

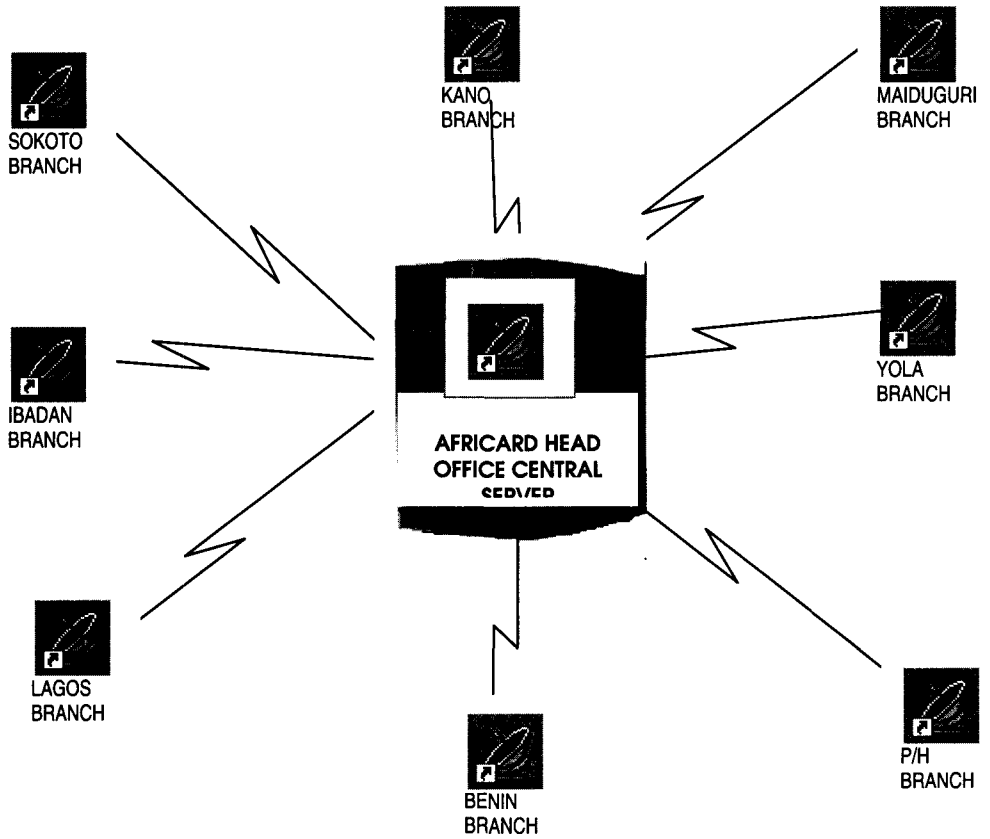
- **Server** (A central server to be located at the Head Office with branches connected to the server online real time).
  - Pentium Server (e.g. Pentium IV 1.5MHz)
  - Sufficient Random Access Memory (RAM) for serving the network
  - Large and fast hard disk (e.g. 30GB)
  - Network card
  - Operating system: Windows 2000 Server
  - Africard E-Purse Software
  - Database: Oracle 9i
  - Backup Device (e.g. tape drive)
  - CD-ROM drive

- **Workstation** (Desktop Computer for each of designated branches)
  - Pentium PC (e.g. Pentium III 1.2GHz)
  - 256MB Random Access Memory (RAM)
  - 20GB hard disk
  - Network card
  - Operating system: Windows 2000 or XP Professional
  - Africard E-Purse Software
  - CD-ROM drive and sound card

- **Point of Sales (POS) machine**
- **Printers (LaserJet)**
- **Uninterrupted Power Supply (UPS)**
- **Stabilizer**
- **Wide Area Network**

4.5 Africard Communication Links

Designated branches for Africard are linked up to the Head Office Africard Server through VSAT.



A central server at the head office with branches connected through VSAT

4.6 Maintenance

For a smart card project to survive, a good maintenance framework must be put in place and sustained.

Foremost, trained personnel must be stationed at all the sub - stations of the project network to attend to errors on the supporting equipment.

All the supporting equipment such as encoder and decoder, servers and energy source must be placed on routine check, to ensure that they are on standby round the clock.

A smart card project, whether loaded or unloaded must be kept save in a manner that all financial transactional instruments are kept under lock and key. It must not be defaced in any form otherwise, the security print within it may be altered. Balance on a smart card must be confirmed periodically to avoid NILL balance.

Loss of smart card must be promptly reported to the issuing authority.

4.7 Training

There will be training on the use of Africard E-purse system, BTT, POS for the Bank's Tellers, POS for the Merchant and all above plus SQL 2000 Server, Windows 2000 Server for the Africard Administrator.

4.8 Starting up the system

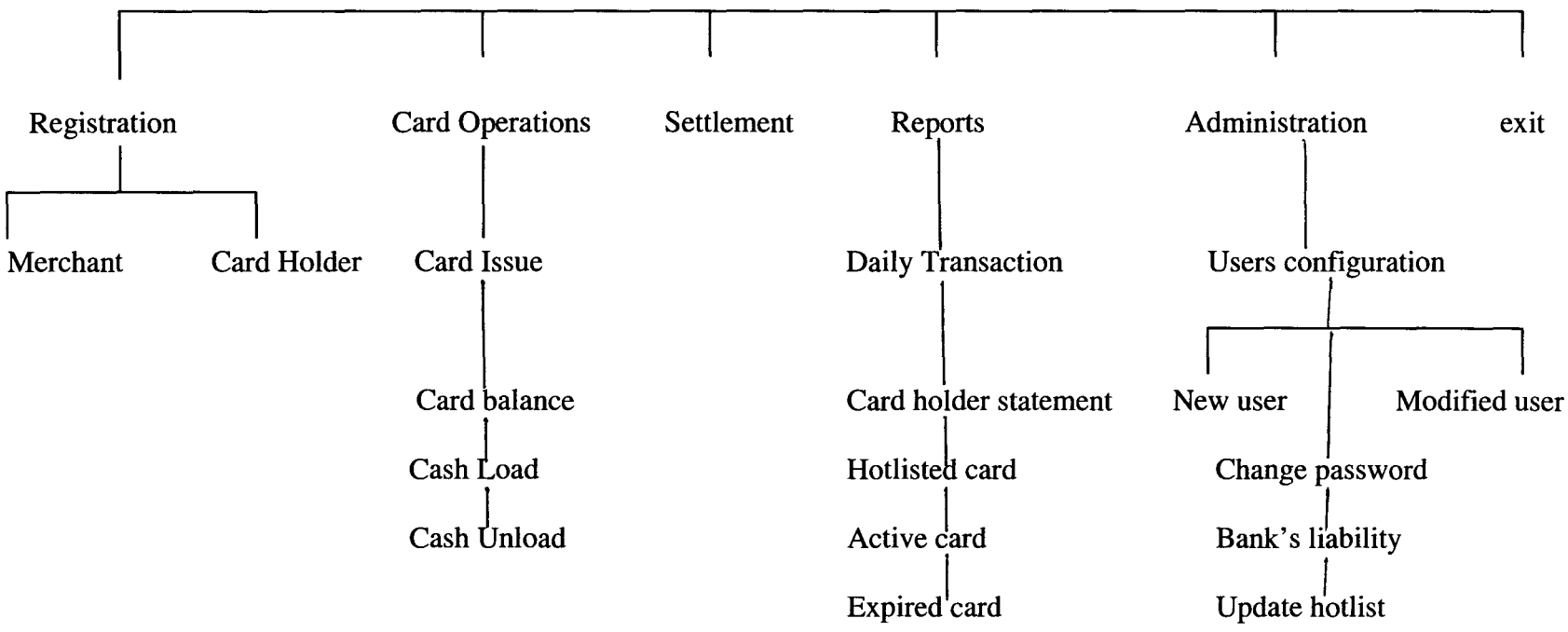
When the system has been successfully installed, the system name Electronic Purse appears on the start menu when clicked upon the main menu comes up after the logging has been successfully recognized by a user with a

valid user identification and password. Once the main menu is on, other outputs are ready for functions.

4.9 Main Menu Structure.

See the attached Diagram

4.9 **Main Menu Structure**



Chapter 5

5.0 Summary

The worldwide growth in the popularity of smart card solutions is driven by the ability of smart cards (e-purse) to securely store information—for payment and other sensitive applications in a variety of environments. This security, provided by the microprocessor chip integrated into a smart card, is opening many application opportunities that were previously impractical or impossible to implement using conventional magnetic-stripe cards.

- Smart cards (e-purse) are platform for lucrative frequent-shopper loyalty programs used either independently (off line) or as integrated components of a retailer's computer system.
- Smart cards which serve as an electronic purse, acting as an electronic payment alternative to bank notes. Automated currency solutions offer greater levels of convenience to consumers, incremental sales opportunities and reduced operating costs to merchants and new service revenue streams to banks.

- Smart card applications reduce operating costs and control fraud in electronic benefits programs, carrying patient information for healthcare applications and providing a secure vehicle for delivering government benefits such as social insurance and welfare programs.
- Smart cards (e-purse) hold the promise of providing secure access to payment and other banking services from the home and other "virtual" environments, allowing consumers to securely initiate payment for goods and services over the Internet and enabling them to reload their stored value electronic purse cards almost anywhere.

Smart cards (e-purse) are used in a variety of other applications that require greater processing power and/or more secure storage. College students use smart cards (e-purse) to pay for cafeteria and bookstore purchases and to access health, recreation and other services; commuters use smart cards (e-purse) to pay tolls and parking fees; and parents use smart cards (e-purse) to pay for child care.

As the number of smart card (e-purse) applications grows, consumer awareness of the convenience provided by smart

card-based applications is driving the adoption of new ways to conduct business. This familiarity increases interest and further accelerates the expansion of opportunities to apply smart card technology.

The growth in the number and variety of smart card applications is also being accelerated by the current flurry of activity within business and industry as banks, credit card companies, equipment manufacturers, processors and others increase their commitment to the use of smart cards (e-purse). This commitment is perhaps best exemplified by the number of partnerships and industry forums devoted to the development and promotion of smart card technology.

5.1 CONCLUSION

In conclusion, all signs point to the rapidly expanded application of smart card technology. Industry leaders are positioning themselves now to profit from the application of this technology through various partnerships and alliances with organizations that are the innovators and principal drivers in the use of smart cards (e-purse).

REFERENCES

1. ALAN DANIELS & DON YEATES, 1989, **Basic Systems Analysis**, 3rd edition, Singapore, ELBS.

2. FERNANDO G. GUENENO & CARLOS DUARO ROJAS, 2001, **SQL Server 2000 Programming by Example**, 1st edition, New York, Que.

3. FRANCESESCO BAKNA, 1999, **Programming Visual Basic 6**, 2nd edition, Microsoft Press.

4. GERALD STUBER, **The Electronic Purse - An Overview of Recent Developments and Policy Issues**, http://www.bankofcanada.ca/

5. JOHN CONNELL, 1998, **Beginning Visual Basic 6 Database Programming**, 1st edition, New Delhi, Wrox Press Inc.

6. JOHN WENNINGER & DAVID LASTER, **The Electronic Purse**, http://www.ny.frb.org/

7. MICHAEL GELLIS, 1999, **Visual Basic 6 Distributed Applications**, New York, Sybex.

8. RICHARD WAYMIRE & RICK SAWTELL, **Sams Teach Yourself Microsoft SQL Server 2000 in 21 Days**, 2nd edition, New York, Sams Publishing.

9. http://www.afribank.com

10. http://www.cenbank.org/

11. http://www.fsbint.com/

12. http://www.smartcardalliance.org/

13. http://www.smartcard.co.uk/

14. http://www.smartcardclub.co.uk/

15. http://www.valucardnigeria.com/

PROGRAM LISTING

APPENDIX

PROGRAM LISTING

PROGRAM LISTING

frmLogin

```
Option Explicit
Private mConn As Connection
Public LoginSucceeded As Boolean
'Public newpassword

Private Sub cmdCancel_Click()
   'set the global var to false
   'to denote a failed login
   LoginSucceeded = False
   Unload Me
End Sub

Private Sub cmdLogin_Click()
If Len(txtUserId.Text) = 0 Or Len(txtPassword.Text) = 0 Then
  MsgBox "Please fill all the boxes...", vbCritical
  txtUserId.SetFocus
  Exit Sub
End If
If Len(txtUserId.Text) < 6 Then
     MsgBox "User Id must be 6 to 12 Characters", vbInformation
     txtUserId.SetFocus
Exit Sub
End If

If Len(txtPassword.Text) < 6 Then
     MsgBox "Password must be 6 to 12 Characters", vbInformation
     txtPassword.SetFocus
Exit Sub
End If
Dim rs As Recordset
 Set rs = New Recordset
 txtUserId.Text = UCase(txtUserId.Text)
 rs.Open "select * from User_tab where User_id = '" + txtUserId.Text + "'",
mConn, adOpenForwardOnly, adLockReadOnly
 If rs.EOF Then
   MsgBox "User Unknown, Contact your Administrator ", vbInformation
 Else
    If Len(rs!user_pass) = 0 Then
      txtPassword.Text = ""
      Load frmNewUser
```

```
      frmNewUser.Show
      Exit Sub
    End If

    If UCase(rs!user_pass) <> UCase(txtPassword.Text)       Then
      MsgBox "Wrong Password ....", vbInformation
    Else

    If rs!start_date_profile > Date Then
      MsgBox "User Profile Start on " & rs!start_date_profile, vbCritical
      Unload Me
      Exit Sub
    End If
    If rs!end_date_profile < Date Then
      MsgBox "User Profile Expired on " & rs!end_date_profile, vbCritical
      Unload Me
      Exit Sub
    End If

 'Load menu & submenus
    Dim rsm As Recordset
    Set rsm = New Recordset
    txtUserId.Text = UCase(txtUserId.Text)
    rsm.Open "select * from epursemenu where Userid = '" + txtUserId.Text +
"'", mConn, adOpenForwardOnly, adLockReadOnly

If Not rsm.EOF Then
 mnuMainMenu.mnuAdministration.Visible = True
 mnuMainMenu.mnuLiability.Enabled = rsm!liability
 mnuMainMenu.mnuChangePassword.Enabled = rsm!changepassword
 mnuMainMenu.mnuModifyUser.Enabled = rsm!modifyuser
 mnuMainMenu.mnuNewUser.Enabled = rsm!newuser
 mnuMainMenu.mnuPasswordReset.Enabled = rsm!passwordreset
 mnuMainMenu.mnuUsersConfig.Enabled = rsm!usersconfig
 mnuMainMenu.mnuUpdateHotlist.Enabled = rsm!updatehotlist

 mnuMainMenu.mnuOperations.Visible = True
 mnuMainMenu.mnuCardBalance.Enabled = rsm!cardbalance
 mnuMainMenu.mnuCardIssue.Enabled = rsm!cardissue
 mnuMainMenu.mnuCashLoad.Enabled = rsm!cashload
 mnuMainMenu.mnuChangePin.Enabled = rsm!changepin
 mnuMainMenu.mnuCashUnload.Enabled = rsm!cashunload
 mnuMainMenu.mnuDownloadHotlist.Enabled = rsm!downloadhotlist
 mnuMainMenu.mnuUploadBTT.Enabled = rsm!uploadbtt
 mnuMainMenu.mnuUploadCard.Enabled = rsm!uploadcard
 mnuMainMenu.mnuUnlockPin.Enabled = rsm!unlockpin
```

```
 mnuMainMenu.mnuRegistration.Visible = True
 mnuMainMenu.mnuCardHolder.Enabled = rsm!cardholder
 mnuMainMenu.mnuMerchantBranch.Enabled = rsm!merchantbranch

 mnuMainMenu.mnuReports.Visible = True
 mnuMainMenu.mnuCardHolderStmt.Enabled = rsm!cardholderstmt

 mnuMainMenu.mnuReports.Visible = True
 mnuMainMenu.mnuDailySettlement.Enabled = rsm!dailysettlement
 'mnuMainMenu.mnuMerchantReport.Enabled = False

 mnuMainMenu.mnuSettlement.Visible = True
 mnuMainMenu.mnuTprocessing.Enabled = rsm!Tprocessing


mnuMainMenu.Show

Else
  MsgBox "Menus not assigned, contact your Administrator", vbCritical
  Unload Me
  Exit Sub
End If

       Unload Me

      Exit Sub
    End If
 End If

 rsm.Close
 Set rsm = Nothing
 Unload Me
End Sub

'Private Sub cmdNewUSer_Click()
'  frmNewUser.Show
'End Sub

Private Sub Form_Load()
 Dim rs As Recordset
 Set mConn = New Connection
  mConn.Open "dsn=epurseodbc;User ID=sa"

End Sub
Private Sub Form_Unload(Cancel As Integer)
```

```
  mConn.Close
  Set mConn = Nothing
End Sub
```

mnuMainMenu

```
Private Sub mnuActiveCards_Click()
frmCRActive.Show
End Sub

Private Sub mnuCardBalance_Click()
 frmCardBalance.Show
End Sub

Private Sub mnuCardHolder_Click()
 frmCardholderData.Show
End Sub

Private Sub mnuCardHolderStmt_Click()
frmCRStmt.Show
End Sub

Private Sub mnuCardIssue_Click()
 frmCardIssue.Show
End Sub

Private Sub mnuCashLoad_Click()
 frmCashLoad.Show
End Sub

Private Sub mnuCashUnload_Click()
 frmCashUnload.Show
End Sub

Private Sub mnuChangePin_Click()
 frmChangePIN.Show
End Sub

Private Sub mnuDailySettlement_Click()
frmCRDaily.Show
End Sub

Private Sub mnuDownloadHotlist_Click()
 frmHotlistDownload.Show
```

```
End Sub

Private Sub mnuExit_Click()
  Unload Me
End Sub

Private Sub mnuExpiredCards_Click()
frmCRExpiredCards.Show
End Sub

Private Sub mnuHotlisted_Click()
frmCRHotlisted.Show
End Sub

Private Sub mnuLiability_Click()
 frmBankLiability.Show
End Sub

Private Sub mnuMenuConfig_Click()
frmEpurseMenu.Show
End Sub

Private Sub mnuMerchantBranch_Click()
 frmMerchantBranch.Show
End Sub

Private Sub mnuModifyUser_Click()
frmModify.Show
End Sub

Private Sub mnuNewUser_Click()
 frmAddUser.Show

End Sub

Private Sub mnuPasswordReset_Click()
 frmPasswordReset.Show
End Sub

Private Sub mnuUnlockPin_Click()
 frmUnlockPIN.Show
End Sub

Private Sub mnuUpdateHotlist_Click()
frmHotlistUpdate.Show
End Sub
```

```
Private Sub munModifyUser_Click()
 frmModify.Show
End Sub

Private Sub mnuUploadBTT_Click()
 frmUploadBTT.Show
End Sub

Private Sub mnuUploadCard_Click()
 frmUploadCard.Show
End Sub
```

frmBankLiability

```
Option Explicit
Private mConn As Connection
Dim rs As Recordset
Private Sub cmdExit_Click()
Unload Me
End Sub

Private Sub cmdSave_Click()
If IsNumeric(txtLiabilityAmount) Then
   rs!liabilityamount = rs!liabilityamount - Val(txtLiabilityAmount)
   rs!AmendmentDate = txtAmendmentDate
   rs.Update
   MsgBox "Done ", vbInformation
   Unload Me
Else
   MsgBox "Enter Valid Number", vbCritical
End If
End Sub

Private Sub Form_Load()
Set mConn = New Connection
mConn.Open "dsn=epurseodbc;User ID=sa"
Set rs = New Recordset
rs.Open "bankliability", mConn, adOpenKeyset, adLockOptimistic
txtAmendmentDate = Date
Label3.Caption = "Current Liability is =N= " & rs!liabilityamount
End Sub

Private Sub Form_Unload(Cancel As Integer)
rs.Close
Set rs = Nothing
```

```
mConn.Close
Set mConn = Nothing
End Sub
```

frmCardBalance

```
Private Sub cmdBalance_Click()
 txtMessage = "Insert Card.."
 txtMessage.Text = "Verifying Card..."
 txtMessage.Text = "Processing ..."
End Sub

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

frmAddUser

```
Option Explicit
Private mConn As Connection
Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdCommit_Click()
On Error GoTo errorhandler
If IsNumeric(txtUserId.Text) Or Len(txtUserId.Text) < 6 Or Len(txtUserId.Text)
> 12 Then
   MsgBox "Enter correct User ID", vbInformation
   txtUserId.SetFocus
   Exit Sub
End If
If IsNumeric(txtUserName.Text) Or Len(txtUserName.Text) = 0 Then
   MsgBox "Enter correct User Name", vbInformation
   txtUserName.SetFocus
   Exit Sub
End If
If Len(txtStaffID.Text) <> 5 Then
   MsgBox "Enter correct Staff ID", vbInformation
   txtStaffID.SetFocus
   Exit Sub
End If
If StartDate.Value < Date Then
   MsgBox "Start Date less than Today", vbInformation
   StartDate.SetFocus
   Exit Sub
```

```
End If
If EndDate.Value < StartDate.Value Then
  MsgBox "End Date less than Start Date", vbInformation
  EndDate.SetFocus
  Exit Sub
End If


 Dim rs As Recordset
 Set rs = New Recordset
 rs.Open "User_tab", mConn, adOpenKeyset, adLockOptimistic

Do While Not rs.EOF
 If UCase(txtUserId.Text) = UCase(rs!user_id) Then
    MsgBox "User Already Exist...", vbCritical
    rs.Close
    Set rs = Nothing
    txtUserId.SetFocus
    Exit Sub
  End If
  rs.MoveNext
Loop


    rs.AddNew
    rs!user_id = UCase(txtUserId.Text)
    rs!user_name = UCase(txtUserName.Text)
    rs!staff_id = UCase(txtStaffID.Text)
    rs!start_date_profile = StartDate.Value
    rs!end_date_profile = EndDate.Value
    rs!user_class = UCase(cboUserClass.Text)
    rs!user_pass = ""
    rs!user_branch = UCase(Branches.Text)
    rs!Remarks = UCase(txtRemarks.Text)
    rs.Update
    rs.Close
    Set rs = Nothing
    MsgBox "User Added Successfully...", vbInformation
    'Exit Sub
    txtUserName.Text = ""
    txtUserId.Text = ""
    txtStaffID.Text = ""
    StartDate.Value = Date
    EndDate.Value = Date
    'cboUserClass.Clear
    Branches.Text = ""
```

```
    txtRemarks.Text = ""
    Exit Sub
errorhandler:
    MsgBox "Critical Error ! ", vbCritical
    Unload Me
End Sub

Private Sub Form_Load()

 Dim rs As Recordset
 Set mConn = New Connection
 'mConn.Open "Provider=SQLOLEDB.1;Persist Security Info=False;User
ID=sa;Initial Catalog=Epurse;Data Source=softpc04"
  mConn.Open "dsn=epurseodbc;User ID=sa"
 StartDate.Value = Date
 EndDate.Value = Date
End Sub
Private Sub Form_Unload(Cancel As Integer)
  mConn.Close
  Set mConn = Nothing
End Sub
```

frmCardHolder

```
Option Explicit
Private mConn As Connection
Dim rs As Recordset


Private Sub cmdExit_Click()
  Unload Me
End Sub

Private Sub cmdSave_Click()
If Len(txtFirstName) = 0 Then
  MsgBox "Enter First Name ", vbCritical
  txtFirstName.SetFocus

  Exit Sub
End If
If Len(txtLastName) = 0 Then
  MsgBox "Enter Last Name ", vbCritical
  txtLastName.SetFocus
  Exit Sub
End If

If EffectiveDate.Value < Date Then
```

```
 MsgBox "Effective Date can't be less than today ", vbCritical
 EffectiveDate.SetFocus

 Exit Sub
End If

If Len(txtMaxBalance) = 0 Or Val(txtMaxBalance) <= 0 Or Val(txtMaxBalance)
> 16000000 Then
 MsgBox "Value should be in the range of =N=1 and =N=16Million", vbCritical
 txtMaxBalance.SetFocus

Exit Sub
End If

 rs.AddNew
 rs!Title = UCase(txtTitle)
 rs!firstname = UCase(txtFirstName)
 rs!middlename = UCase(txtMiddleName)
 rs!lastname = UCase(txtLastName)
 rs!idtype = UCase(txtIDType)
 rs!idstring = UCase(txtIDString)
 rs!language = UCase(cboLanguage)
 rs!account = txtAccount
 rs!EffectiveDate = EffectiveDate
 rs!expirydate = txtExpiryDate
 rs!cardnumber = txtCardNumber
 rs!ncurrency = txtCurrency
 rs!maxbalance = Val(txtMaxBalance)
 rs!PinBalance = PinBalance
 rs!PinCredit = PinCredit
 rs!PinDebit = PinDebit

 rs.Update

 init1
End Sub
Private Sub init1()
txtTitle = ""
txtFirstName = ""
txtMiddleName = ""
txtLastName = ""
txtIDType = ""
txtIDString = ""
cboLanguage = "English"
txtAccount = ""
EffectiveDate = Date
```

```
txtCardNumber = rs!cardnumber + 1
PinCredit = 0
txtTitle.SetFocus

End Sub
Private Sub EffectiveDate_Change()
txtExpiryDate = Format(EffectiveDate.Value + 730, "dd/mm/yyyy")

End Sub

Private Sub Form_Load()
 EffectiveDate.Value = Date
 txtExpiryDate = Format(EffectiveDate.Value + 730, "dd/mm/yyyy")

 Set mConn = New Connection
 mConn.Open "dsn=epurseodbc;User ID=sa"
 Set rs = New Recordset
 rs.Open "select * from Cardholderdetails order by cardnumber", mConn,
adOpenKeyset, adLockOptimistic
 rs.MoveLast
 txtCardNumber = rs!cardnumber + 1
End Sub

Private Sub txtFirstName_Change()
txtCommonName = txtTitle & " " & Trim(txtFirstName) & " " &
Trim(txtMiddleName) & " " & Trim(txtLastName)
End Sub

Private Sub txtLastName_Change()
 txtCommonName = txtTitle & " " & Trim(txtFirstName) & " " &
Trim(txtMiddleName) & " " & Trim(txtLastName)

End Sub

Private Sub txtMiddleName_Change()
 txtCommonName = txtTitle & " " & Trim(txtFirstName) & " " &
Trim(txtMiddleName) & " " & Trim(txtLastName)

End Sub
Private Sub Form_Unload(Cancel As Integer)
  rs.Close
  Set rs = Nothing
  mConn.Close
  Set mConn = Nothing
End Sub
```

frmCardIssue

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

Private Sub cmdIssue_Click()
txtMessage = "Verifying...."
txtMessage = "Enter PIN.."
FF = InputBox(g, "ENTER PIN")
'txtMessage = "Confirm PIN..."
JJ = InputBox(I, "CONFIRM PIN")
If JJ <> FF Then
  MsgBox "PIN not the same", vbCritical
  txtMessage = "Try Again"
Else
  txtMessage = "Issued Successfully.."
End If
End Sub
```

frmCashLoad

```
Option Explicit
Private mConn As Connection
Dim rs As Recordset
Dim rsb As Recordset
Private Sub cmdAccept_Click()
Dim result As Currency
If IsNumeric(txtAmount) And Val(txtAmount) > 0 Then
   If Val(txtAmount) > Abs(rsb!liabilityamount) Then
      MsgBox "Current Liability Exceeded, contact System Administrator",
vbInformation
      Exit Sub
    Else
     txtMessage.Text = "Verifying Card..."
     txtMessage.Text = "Processing ..."

     txtMessage.Text = "Cash Load Successful"
     lblbranch = "Broad Street Branch"
     lbldatetime = Now
     lblcashload = "CASH LOAD"
     lblsign = "CREDIT =N= "
     lblamount = txtAmount
     rs.AddNew
     rs!transactionid = Format(Now, "ddmmyyyy") & "/" & Format(Now,
"hhmmss")
```

```
    rs!transactiontype = "CASH LOAD"
    rs!cardnumber = lblcardnumber
    rs!catid = lblcatid
    rs!caid = lblcaid
    rs!DateTime = Now
    rs!amount = txtAmount
    rs.Update
    result = rsb!liabilityamount + Val(txtAmount)
    rsb!liabilityamount = result
    rsb.Update
    MsgBox "Cash Load Successful..", vbInformation
    End If
   Else
     MsgBox "Enter Correct Value ..", vbCritical
     txtAmount = ""
     txtAmount.SetFocus
   End If
End Sub

Private Sub cmdExit_Click()
 Unload Me
 End Sub

Private Sub Form_Load()
Set mConn = New Connection
mConn.Open "dsn=epurseodbc;User ID=sa"
Set rs = New Recordset
rs.Open "transactiondaily", mConn, adOpenKeyset, adLockOptimistic
Set rsb = New Recordset
rsb.Open "bankliability", mConn, adOpenKeyset, adLockOptimistic
End Sub

Private Sub Form_Unload(Cancel As Integer)
rs.Close
Set rs = Nothing
rsb.Close
Set rsb = Nothing
mConn.Close
Set mConn = Nothing
End Sub
```

frmCashUnload

```
Option Explicit
Private mConn As Connection
Dim rs As Recordset
```

```
Dim rsb As Recordset
Private Sub cmdAccept_Click()
Dim result As Currency
If IsNumeric(txtAmount) And Val(txtAmount) > 0 Then

    txtMessage.Text = "Processing ..."
    txtMessage.Text = "Enter PIN..."

    txtMessage.Text = "Cash Unload Successful"
    lbldatetime = Now
    lblcashload = "CASH UNLOAD"
    lblsign = "DEBIT =N= "
    lblamount = txtAmount
    rs.AddNew
    rs!transactionid = Format(Now, "ddmmyyyy") & "/" & Format(Now,
"hhmmss")
    rs!transactiontype = "CASH UNLOAD"
    rs!cardnumber = lblcardnumber
    rs!catid = lblcatid
    rs!caid = lblcaid
    rs!DateTime = Now
    rs!amount = txtAmount
    rs.Update
    result = rsb!liabilityamount - Val(txtAmount)
    rsb!liabilityamount = result
    rsb.Update
    MsgBox "Cash Unload Successful..", vbInformation

   Else
     MsgBox "Enter Correct Value ..", vbCritical
     txtAmount = ""
     txtAmount.SetFocus
   End If

End Sub

Private Sub cmdExit_Click()
 Unload Me
 End Sub

Private Sub Form_Load()
Set mConn = New Connection
mConn.Open "dsn=epurseodbc;User ID=sa"
Set rs = New Recordset
rs.Open "transactiondaily", mConn, adOpenKeyset, adLockOptimistic
Set rsb = New Recordset
```

```
rsb.Open "bankliability", mConn, adOpenKeyset, adLockOptimistic
End Sub
Private Sub Form_Unload(Cancel As Integer)
rs.Close
Set rs = Nothing
rsb.Close
Set rsb = Nothing
mConn.Close
Set mConn = Nothing
End Sub
```

frmChangePIN

```
Private Sub cmdChangePIN_Click()
txtMessage = "Verifying...."
'txtMessage = "Enter old PIN.."
FF = InputBox(g, "ENTER OLD PIN")
'txtMessage = "Confirm PIN..."
JJ = InputBox(I, "ENTER NEW PIN")
'If JJ <> FF Then
 ' MsgBox "PIN not the same", vbCritical
  'txtMessage = "Try Again"
'Else
  txtMessage = "PIN Changed Successfully.."
End Sub

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

frmEpurseMenu

```
Option Explicit
Private mConn As Connection

Private Sub cboUserId_Click(Area As Integer)
If Len(cboUserId.Text) = 0 Then
  MsgBox "Select Valid User ", vbCritical
  cboUserId.SetFocus
  Exit Sub
End If
 Dim rs As Recordset
 Set rs = New Recordset
 rs.Open "select * from epursemenu where userid = '" & cboUserId & "'",
mConn, adOpenForwardOnly, adLockReadOnly
If Not rs.EOF Then
' cboUserId.Text = rs!userid
```

```
 'Adm = rs!administration
 AdmBl = rs!liability
 AdmCp = rs!changepassword
 AdmMu = rs!changepassword
 AdmNu = rs!newuser
 AdmPr = rs!passwordreset
 AdmUc = rs!usersconfig
 AdmUh = rs!updatehotlist
 AdmMc = rs!menuconfig
 'Ops = rs!operations
 OpsCb = rs!cardbalance
 OpsCi = rs!cardissue
 OpsCl = rs!cashload
 OpsCp = rs!changepin
 OpsCu = rs!cashunload
  OpsDh = rs!downloadhotlist
 OpsUb = rs!uploadbtt
 OpsUc = rs!uploadcard
 OpsUp = rs!unlockpin
 'Reg = rs!registration
 RegCh = rs!cardholder
 RegMb = rs!merchantbranch
' Rep = rs!reports
 RepCs = rs!cardholderstmt
  RepDs = rs!dailysettlement
 RepMr = rs!merchantreport
 'Sett = rs!settlement
 SettTp = rs!Tprocessing
End If
 rs.Close
 Set rs = Nothing
End Sub

Private Sub cmdExit_Click()
 Unload Me
End Sub

Private Sub cmdSave_Click()
If Len(cboUserId.Text) = 0 Then
  MsgBox "Select Valid User ", vbCritical
  cboUserId.SetFocus
  Exit Sub
End If
Dim rs As Recordset
 Set rs = New Recordset
```

```
 rs.Open "select * from epursemenu where userid = '" & cboUserId & "'",
mConn, adOpenKeyset, adLockOptimistic
 If rs.EOF Then
   rs.AddNew
 End If

  rs!userid = cboUserId.Text
' rs!administration = Adm
 rs!liability = AdmBl
 rs!changepassword = AdmCp
 rs!modifyuser = AdmMu
 rs!newuser = AdmNu
 rs!passwordreset = AdmPr
 rs!usersconfig = AdmUc
 rs!updatehotlist = AdmUh
 rs!menuconfig = AdmMc
 'rs!operations = Ops
 rs!cardbalance = OpsCb
 rs!cardissue = OpsCi
 rs!cashload = OpsCl
 rs!changepin = OpsCp
 rs!cashunload = OpsCu
 rs!downloadhotlist = OpsDh
 rs!uploadbtt = OpsUb
 rs!uploadcard = OpsUc
 rs!unlockpin = OpsUp
 'rs!registration = Reg
 rs!cardholder = RegCh
 rs!merchantbranch = RegMb
 'rs!reports = Rep
 rs!cardholderstmt = RepCs
 rs!dailysettlement = RepDs
 rs!merchantreport = RepMr
 'rs!settlement = Sett
 rs!Tprocessing = SettTp.Value

 rs.Update
 rs.Close
 Set rs = Nothing
 MsgBox "Menu successfully allocated ", vbInformation
End Sub

Private Sub Form_Load()
 Dim rs As Recordset
 Set mConn = New Connection
 mConn.Open "dsn=epurseodbc;User ID=sa"
```

```
End Sub

Private Sub Form_Unload(Cancel As Integer)
 mConn.Close
 Set mConn = Nothing
End Sub
```

frmHotlistUpdate

```
Option Explicit
Private mConn As Connection
Dim rs As Recordset

Private Sub cmdExit_Click()
  Unload Me
End Sub

Private Sub cmdDefault_Click()
 Unload Me
End Sub

Private Sub cmdUpdate_Click()
If Len(txtCardNumber) = 0 Then
  MsgBox "Enter Correct Card Number ", vbCritical
  txtCardNumber.SetFocus
  Exit Sub
End If
Set rs = New Recordset
 rs.Open "select * from Cardholderdetails where cardnumber = '" &
txtCardNumber & "'", mConn, adOpenKeyset, adLockOptimistic
If rs.EOF Then
  MsgBox txtCardNumber & " does not exist ", vbCritical
  txtCardNumber.SetFocus
  Exit Sub
Else
  rs!hotlist = "Y"
  rs.Update
End If
 rs.Close
  Set rs = Nothing
  MsgBox "Card Hotlisted...", vbInformation
End Sub

Private Sub Form_Load()
```

```
 Set mConn = New Connection
 mConn.Open "dsn=epurseodbc;User ID=sa"


End Sub

Private Sub Form_Unload(Cancel As Integer)

  mConn.Close
  Set mConn = Nothing
End Sub
```

frmMerchantBranchName

```
Option Explicit
Private mConn As Connection
Dim rs As Recordset
Private Sub cmdExit_Click()
 Unload Me
End Sub

Private Sub cmdSave_Click()
If Len(txtBranchMerchantName) = 0 Then
  MsgBox "Enter Branch/Merchant Name ", vbCritical
  txtBranchMerchantName.SetFocus
  Exit Sub
End If

If Len(txtAddress) = 0 Then
  MsgBox "Enter Address ", vbCritical
  txtAddress.SetFocus
  Exit Sub
End If
If Len(txtCatid) = 0 Then
  MsgBox "Enter CATID ", vbCritical
  txtCatid.SetFocus
  Exit Sub
End If
If Len(txtCaid) = 0 Then
  MsgBox "Enter CAID ", vbCritical
  txtCaid.SetFocus
  Exit Sub
End If
rs.AddNew
rs!branchmerchantname = UCase(txtBranchMerchantName)
rs!address = UCase(txtAddress)
rs!catid = UCase(txtCatid)
```

```
rs!caid = UCase(txtCaid)
rs.Update
txtBranchMerchantName = ""
txtAddress = ""
txtCatid = ""
txtCaid = ""
txtBranchMerchantName.SetFocus
End Sub

Private Sub Form_Load()
 Set mConn = New Connection
 mConn.Open "dsn=epurseodbc;User ID=sa"
 Set rs = New Recordset
 rs.Open "branchmerchant", mConn, adOpenKeyset, adLockOptimistic
End Sub

Private Sub Form_Unload(Cancel As Integer)
rs.Close
Set rs = Nothing
mConn.Close
Set mConn = Nothing
End Sub
```

frmModify

```
Option Explicit
Private mConn As Connection

Private Sub cmdClose_Click()
Unload Me
End Sub

Private Sub cmdCommit_Click()
If IsNumeric(txtUserName.Text) Or Len(txtUserName.Text) = 0 Then
   MsgBox "Enter correct User Name", vbInformation
   txtUserName.SetFocus
   Exit Sub
End If
If Len(txtStaffID.Text) <> 5 Then
   MsgBox "Enter correct Staff ID", vbInformation
   txtStaffID.SetFocus
   Exit Sub
End If
If StartDate.Value < Date Then
   MsgBox "Start Date less than Today", vbInformation
   StartDate.SetFocus
```

```
    Exit Sub
End If
If EndDate.Value < StartDate.Value Then
    MsgBox "End Date less than Start Date", vbInformation
    EndDate.SetFocus
    Exit Sub
End If

dbUserId.Enabled = False

Dim rs As Recordset
 Set rs = New Recordset
 rs.Open "User_tab", mConn, adOpenKeyset, adLockOptimistic
 Do While Not rs.EOF
  If UCase(dbUserId.Text) = UCase(rs!user_id) Then
     'rs!user_id = UCase(txtUserId.Text)
     rs!user_name = UCase(txtUserName.Text)
     rs!staff_id = UCase(txtStaffID.Text)
     rs!start_date_profile = StartDate.Value
     rs!end_date_profile = EndDate.Value
     rs!user_class = UCase(cboUserClass.Text)

     'rs!user_pass = rs!user_id
     rs!user_branch = UCase(Branches.Text)
     rs!Remarks = UCase(txtRemarks.Text)
     rs.Update
     MsgBox "User Successfully Modified...", vbInformation
     dbUserId.Enabled = True
     initspace
     initfalse
     Exit Sub
    End If
    rs.MoveNext
  Loop
  rs.Close
  Set rs = Nothing


End Sub

Private Sub cmdDelete_Click()
On Error GoTo errorhandler

dbUserId.Enabled = False
Dim rs As Recordset
 Set rs = New Recordset
```

```
 rs.Open "User_tab", mConn, adOpenKeyset, adLockOptimistic
 Do While Not rs.EOF
  If UCase(dbUserId.Text) = UCase(rs!user_id) Then
    rs.Delete
    MsgBox "User Successfully Delete ...", vbInformation
    initspace
    initfalse
    dbUserId.Enabled = True

    Exit Sub
  End If
 rs.MoveNext
 Loop
 'Adodc2.Refresh
 rs.Close
 Set rs = Nothing
Exit Sub
errorhandler:
    MsgBox "Critical Error ! ", vbCritical
    Unload Me
End Sub

Private Sub cmdRetrieve_Click()
 Dim rs As Recordset
 Set rs = New Recordset
  rs.Open "user_tab", mConn, adOpenForwardOnly, adLockReadOnly

 Do While Not rs.EOF
  If UCase(dbUserId.Text) = UCase(rs!user_id) Then
   inittrue
    txtUserName.Text = rs!user_name
    txtStaffID.Text = rs!staff_id
    StartDate.Value = rs!start_date_profile
    EndDate.Value = rs!end_date_profile
    cboUserClass.Text = rs!user_class
    Branches.Text = rs!user_branch
    txtRemarks.Text = rs!Remarks
   rs.Close
   Set rs = Nothing

   Exit Sub
  End If
  rs.MoveNext
Loop

  If rs.EOF Then
```

```
  MsgBox "User Not Found...", vbCritical
  End If
End Sub

Private Sub DataCombo1_Click(Area As Integer)

End Sub

Private Sub dbUserId_Click(Area As Integer)
'Adodc2.Refresh
Dim rs As Recordset
 Set rs = New Recordset
  rs.Open "user_tab", mConn, adOpenForwardOnly, adLockReadOnly

 Do While Not rs.EOF
  If UCase(dbUserId.Text) = UCase(rs!user_id) Then
   inittrue
     txtUserName.Text = rs!user_name
     txtStaffID.Text = rs!staff_id
     StartDate.Value = rs!start_date_profile
     EndDate.Value = rs!end_date_profile
     cboUserClass.Text = rs!user_class
     Branches.Text = rs!user_branch
     txtRemarks.Text = rs!Remarks
    rs.Close
    Set rs = Nothing

   Exit Sub
  End If
  rs.MoveNext
Loop

'  If rs.EOF Then
 ' MsgBox "User Not Found...", vbCritical
 ' txtUserId.SetFocus
  'End If
End Sub

Private Sub Form_Load()
  StartDate.Value = Date
  EndDate.Value = Date
  initfalse
  Dim rs As Recordset
 Set mConn = New Connection
 'mConn.Open "Provider=SQLOLEDB.1;Persist Security Info=False;User
ID=sa;Initial Catalog=epurse;Data Source=softpc04"
```

```
  mConn.Open "dsn=epurseodbc;User ID=sa"

End Sub
Private Sub initfalse()
    txtUserName.Enabled = False
    'txtUserId.Enabled = False
    txtStaffID.Enabled = False
    StartDate.Enabled = False
    EndDate.Enabled = False
    cboUserClass.Enabled = False
    Branches.Enabled = False
    txtRemarks.Enabled = False
    cmdDelete.Enabled = False
    cmdCommit.Enabled = False
End Sub

Private Sub inittrue()
    txtUserName.Enabled = True
    'txtUserId.Enabled = true
    txtStaffID.Enabled = True
    StartDate.Enabled = True
    EndDate.Enabled = True
    cboUserClass.Enabled = True
    Branches.Enabled = True
    txtRemarks.Enabled = True
    cmdDelete.Enabled = True
    cmdCommit.Enabled = True
End Sub
Private Sub initspace()
   txtUserName.Text = ""
    'txtUserId.Text = ""
    txtStaffID.Text = ""
    StartDate.Value = Date
    EndDate.Value = Date
    'cboUserClass.Clear
    Branches.Text = ""
    txtRemarks.Text = ""
End Sub
Private Sub Form_Unload(Cancel As Integer)
  mConn.Close
  Set mConn = Nothing
End Sub
```

frmNewUser

```
Option Explicit
```

```
Private mConn As Connection
Public LoginSucceeded As Boolean

Private Sub cmdCancel_Click()
  Unload Me
End Sub

Private Sub cmdLogin_Click()

If Len(txtUserId.Text) = 0 Or Len(txtPassword.Text) = 0 Or
Len(txtRepeatPass.Text) = 0 Then
   MsgBox "Please fill all the boxes...", vbCritical
   txtPassword.SetFocus
   Exit Sub
End If
If Len(txtPassword.Text) < 6 Then
      MsgBox "Password must be 6 to 12 Characters", vbInformation
      txtPassword.SetFocus
   Exit Sub
End If

Dim rs As Recordset
 Set rs = New Recordset
 txtUserId.Text = UCase(txtUserId.Text)
 rs.Open "select * from User_tab where User_id = '" + txtUserId.Text + "'",
mConn, adOpenKeyset, adLockOptimistic
 If rs.EOF Then
   MsgBox "User Unknown, Contact your Administrator ", vbInformation
   Unload Me
 Else
    If Len(rs!user_pass) <> 0 Then
       MsgBox "You are not a new user....", vbInformation
       Load frmNewUser
       frmNewUser.Show
       Exit Sub
       Unload Me
    Else
     If UCase(txtPassword.Text) <> UCase(txtRepeatPass.Text) Then
        MsgBox "Password not the same ....", vbInformation
        txtPassword.Text = ""
        txtRepeatPass.Text = ""
        txtPassword.SetFocus
        Exit Sub
       'Unload Me
      End If
```

```
      rs!user_pass = UCase(txtPassword.Text)
      rs.Update
      Unload Me
      Exit Sub
    End If
  End If

  rs.Close
  Set rs = Nothing
End Sub

Private Sub Form_Load()
  txtUserId.Text = frmLogin.txtUserId.Text
  txtUserId.Enabled = False
  Dim rs As Recordset
  Set mConn = New Connection
  mConn.Open "dsn=epurseodbc;User ID=sa"

End Sub
Private Sub Form_Unload(Cancel As Integer)
   mConn.Close
   Set mConn = Nothing
End Sub
```

frmPasswordReset

```
Option Explicit
Private mConn As Connection

Private Sub cmdClose_Click()
  Unload Me
End Sub

Private Sub cmdCommit_Click()
On Error GoTo errorhandler
dbUserId.Enabled = False

Dim rs As Recordset
  Set rs = New Recordset
  rs.Open "User_tab", mConn, adOpenKeyset, adLockOptimistic
  Do While Not rs.EOF
   If UCase(dbUserId.Text) = UCase(rs!user_id) Then
     'rs!user_id = UCase(txtUserId.Text)
     rs!user_pass = ""
     rs.Update
```

```
    MsgBox "User Successfully Reset...", vbInformation
    dbUserId.Enabled = True
    initspace
    initfalse
    Exit Sub
   End If
   rs.MoveNext
  Loop
  rs.Close
  Set rs = Nothing
 Exit Sub
errorhandler:
    MsgBox "Critical Error ! ", vbCritical
    Unload Me

End Sub

Private Sub cmdDelete_Click()
dbUserId.Enabled = False
Dim rs As Recordset
 Set rs = New Recordset
 rs.Open "User_tab", mConn, adOpenKeyset, adLockOptimistic
 Do While Not rs.EOF
  If UCase(dbUserId.Text) = UCase(rs!user_id) Then
    rs.Delete
    MsgBox "User Successfully Delete ...", vbInformation
    initspace
    initfalse
    dbUserId.Enabled = True

    Exit Sub
  End If
 rs.MoveNext
 Loop
 rs.Close
 Set rs = Nothing

End Sub



Private Sub dbUserId_Click(Area As Integer)
Dim rs As Recordset
 Set rs = New Recordset
  rs.Open "user_tab", mConn, adOpenForwardOnly, adLockReadOnly
```

```
 Do While Not rs.EOF
  If UCase(dbUserId.Text) = UCase(rs!user_id) Then
   'inittrue
     txtUserName.Text = rs!user_name
     txtStaffID.Text = rs!staff_id
     StartDate.Text = rs!start_date_profile
     EndDate.Text = rs!end_date_profile
     UserClass.Text = rs!user_class
     Branches.Text = rs!user_branch
     txtRemarks.Text = rs!Remarks
     cmdCommit.Enabled = True
    rs.Close
    Set rs = Nothing

   Exit Sub
  End If
  rs.MoveNext
Loop

'  If rs.EOF Then
 ' MsgBox "User Not Found...", vbCritical
 ' txtUserId.SetFocus
  'End If
End Sub

Private Sub Form_Load()

  initfalse
  Dim rs As Recordset
 Set mConn = New Connection
 'mConn.Open "Provider=SQLOLEDB.1;Persist Security Info=False;User
ID=sa;Initial Catalog=epurse;Data Source=softpc04"
  mConn.Open "dsn=epurseodbc;User ID=sa"
End Sub
Private Sub initfalse()
    txtUserName.Enabled = False
    'txtUserId.Enabled = False
    txtStaffID.Enabled = False
    StartDate.Enabled = False
    EndDate.Enabled = False
    UserClass.Enabled = False
    Branches.Enabled = False
    txtRemarks.Enabled = False
    cmdCommit.Enabled = False
End Sub
```

```
Private Sub inittrue()
  txtUserName.Enabled = True
    'txtUserId.Enabled = true
    txtStaffID.Enabled = True
    StartDate.Enabled = True
    EndDate.Enabled = True
    UserClass.Enabled = True
    Branches.Enabled = True
    txtRemarks.Enabled = True
    cmdCommit.Enabled = True
End Sub
Private Sub initspace()
   txtUserName.Text = ""
    'txtUserId.Text = ""
    txtStaffID.Text = ""
    StartDate.Text = ""
    EndDate.Text = ""
    UserClass.Text = ""
    Branches.Text = ""
    txtRemarks.Text = ""
End Sub

Private Sub Form_Unload(Cancel As Integer)
 mConn.Close
 Set mConn = Nothing
End Sub
```

frmUnlockPIN

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

Private Sub cmdUnlockPIN_Click()
txtMessage = "Verifying...."
FF = InputBox(g, "ENTER PIN")

JJ = InputBox(l, "CONFIRM PIN")
If JJ <> FF Then
 MsgBox "PIN not the same", vbCritical
 txtMessage = "Try Again"
Else
 txtMessage = "Unlocked Successfully.."
End If
End Sub
```

```
Private Sub Form_Load()

End Sub
```

frmUploadBTT

```
Option Explicit
Private mConn As Connection
Dim rsm As Recordset
Dim rsd As Recordset


Private Sub cmdExit_Click()
   Unload Me
End Sub

Private Sub cmdUpload_Click()
  Dim rec As Integer
  rec = 0
  Do While Not rsd.EOF
    rsm.AddNew
    rsm!transactionid = rsd!transactionid
    rsm!transactiontype = rsd!transactiontype
    rsm!cardnumber = rsd!cardnumber
    rsm!catid = rsd!catid
    rsm!caid = rsd!caid
    rsm!DateTime = rsd!DateTime
    rsm!amount = rsd!amount
    rsm!userid = rsd!userid
    rsm.Update
    rec = rec + 1
    txtMessage = "Processing Record.." & rec
    rsd.Delete
    rsd.MoveNext
 Loop
    MsgBox "Upload completed...", vbInformation
End Sub

Private Sub Form_Load()
   Set mConn = New Connection
   mConn.Open "dsn=epurseodbc;User ID=sa"
   Set rsm = New Recordset
   rsm.Open "transactionmaster", mConn, adOpenKeyset, adLockOptimistic
   Set rsd = New Recordset
   rsd.Open "transactiondaily", mConn, adOpenKeyset, adLockOptimistic
End Sub
```

```
Private Sub Form_Unload(Cancel As Integer)
   rsm.Close
   Set rsm = Nothing
   rsd.Close
   Set rsd = Nothing
   mConn.Close
   Set mConn = Nothing
End Sub
```

frm

SAMPLE OUTPUT

DAILY TRANSACTION

TRANSACTION ID	CARDNO	ACCEPTOR TERMINAL ID	ACCEPTOR ID	AMOUNT	TYPE	DATE/TIME
100002	100001112	CORII	MUSA23	50000	CR	12/10/2006
100003	100001113	CORIII	DAN25	60000	CR	10/07/2006
100004	100001114	CORIV	ALLI 17	55000	CR	22/08/2006
100005	100001115	COR V	MUSA28	65000	CR	18/09/2006
100006	100001116	CORVI	ISSA15	70000	CR	30/11/2006

CARD HOLDER

CARD NUMBER 100001112 DATE/TIME OF TRANSACTION: 12/10/06

CARD HOLDER NAME: ALI MUSTAPHA ALBERT

TRANSACTION TYPE: CREDIT

BRANCH / MERCHANT NAME: MINNA TERMINAL

DEBIT:0.00 CREDIT: 20000

HOTLISTED CARDS

CARD NUMBER	CARD HOLDER NAME:	HOTLISTED DATE
100001112	ALI MUSTAPHA ALBERT	12/10/2006
100001113	DANLADI YAKUBU	10/07/2006
100001114	COLLINS GEORGE	22/08/2006
100001115	MUSA ABUBAKAR	18/09/2006
100001116	IBRAHEEM SETIMA	30/11/2006

ACTIVE CARDS

CARD NUMBER	CARD HOLDER NAME:	EXPIRY DATE
100001112	ALI MUSTAPHA ALBERT	12/10/2010
100001113	DANLADI YAKUBU	10/07/2010
100001114	COLLINS GEORGE	22/08/2010
100001115	MUSA ABUBAKAR	18/09/2010
100001116	IBRAHEEM SETIMA	30/11/2010

EXPIRY CARDS

CARD NUMBER	CARD HOLDER NAME:	EXPIRY DATE
100001112	ALI MUSTAPHA ALBERT	12/10/2007
100001113	DANLADI YAKUBU	10/07/2007
100001114	COLLINS GEORGE	22/08/2007
100001115	MUSA ABUBAKAR	18/09/2007
100001116	IBRAHEEM SETIMA	30/11/2007