BANK MANAGEMENT SYSTEM

by

NAME Bandaru Kedaarnath	**REGISTER NUMBER** 19BCE1370
NAME Biyyapu SriVardhan Reddy	**REGISTER NUMBER** 19BCE1338
NAME Nagineni Ashish	**REGISTER NUMBER** 19BCE1333
NAME Aayush Kumar Singh	**REGISTER NUMBER** 19BCE1113

A project report submitted to

Dr.FLORENCE GNANA POOVATHY J

SCHOOL OF ELECTRONICS ENGINEERING

in partial fulfilment of the requirements for the course of

CSE2006 – MICROPROCESSORS AND INTERFACING

in

B. Tech. ELECTRONICS AND COMMUNICATION ENGINEERING

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

Vandalur – Kelambakkam Road

Chennai – 600127

JUNE 2021

BONAFIDE CERTIFICATE

Certified that this project report entitled "**BANK MANAGEMENT SYSTEM"** is a bonafide work of **BANDARU KEDAARNATH–19BCE1370, BIYYAPU SRIVARDHAN REDDY-19BCE1338, NAGINENI ASHISH-19BCE1333 and AAYUSH KUMAR SINGH-19BCE1113 who** carried out the Project work under my supervision and guidance for **CSE2006-MICROPROCESSOR AND INTERFACING.**

Dr. FLORENCE GNANA POOVATHY J

Associate Professor Senior Grade 1

School of Electronics Engineering (SENSE),

VIT University, Chennai

Chennai – 600 127.

ABSTRACT

The Bank Account Management System is an application for maintaining a person's account in a bank. In this project we tried to show the working of a banking account system and cover the basic functionality of a Bank Account Management System using assembly language programming (ALP). We develop this project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user's work space to have additional functionalities which are not provided under a conventional banking project. The Bank Account Management System undertaken as a project is based on relevant technologies. This project is developed using Assembly language programming (ALP) and emulator. Creating and managing requirements is a challenge of IT, systems and product development projects or indeed for any activity where you have to manage a contractual relationship. Organization need to effectively define and manage requirements to ensure they are meeting needs of the customer, while proving compliance and staying on the schedule and within budget. The impact of a poorly expressed requirement can bring a business out of compliance or even cause injury or death. Requirements definition and management is an activity that can deliver a high, fast return on investment. The project analyzes the system requirements and then comes up with the requirements specifications. It studies other related systems and then come up with system specifications. The system is designed as an interactive and content management system. The content management system deals with data entry, validation confirm and updating whiles the interactive system deals with system interaction with the administration and users. Thus, above features of this project will save transaction time and therefore increase the efficiency of the system.

TABLE OF CONTENTS

1. INTRODUCTION

Our Project is mainly based on banking management System there has been a major change in the way the banks operate. Banking system changed the system of manual operations significantly to technological operations. These days many banks have switched to net banking, UPI operations, and more operations which are considered to be long taking process in olden days. Various banks which operate agricultural, business, jewellery, money started moving to online systems. This has brought a lot of changes in the user's usage. Microprocessors have revolutionized the computer industry from its beginning. Computer processors have led to various other processors that functions various tasks by reducing time utilization. This is also applicable to the banking industry, where atm machines, banking machines made a lot of work easier. The emulator application is a application which leads to execute various processes using Assembly language, to give a analysis of the atm machines, bank machines, airport banking machines we see today. The application is made using emulator which follows various functions like creating a bank account, deletion of it, showing balance, withdraw and some other processes that we were familiar. The user interface is not the general interface as we see in atm, banking machines but gives a glimpse of what they do. This application is related to the processors, which led to use emulator and assemble programming language for this application. User can experience the various features that were similar to atm machines, but does not link with banks or any other business format, but gives a glimpse of the advanced banking system. This technology advancement have solved various problem of the past. The emulator performs operation as of the machines perform so it has been chosen for this application.

2. MODULAR DESCRIPTION

2.1 MENU PAGE

The banking system application generally has a user interface similar to ATM machines. This application will only represent the use of microprocessors in real life applications. The menu bar will give options for user to go through those and modify their account status regularly. The menu page plays a key role in this application.

The options on Menu Page:

- Create Account.
- Print details of account
- Deposit
- Withdraw
- Reset
- Modify details

2.2 CREATE ACCOUNT

As menu page be an interface between the user and the machine, one of the key features is creating an account. In real life bank create account process, it has to be done face to face but not in virtual mode. Number of documents require to the whole create account process. This application asks users to type their basic details like name, account number, pin. The account creation process be done, but the verification process will be done in another section of this application.

2.3 PRINT ACCOUNT DETAILS

The account printing details is a process which is present in ATM machines. As user withdraws or performs operations the ATM machine will generate a page of details printed about our account, it contains Our name, address, details like balance, time of process, account number. This application has the same feature featuring the print of user's account details in an orderly fractioned manner. User can check and modify details in Modify details section present in this application.

2.4 DEPOSIT

User can deposit money in ATM machines, banking application in airports. They need to deposit through the bank branch, now a days advanced application like UPI's made the process to simple. But this banking system application follows similar way of depositing money but follows process of deposit through bank branches. User needs to enter account details for verification purpose, and then fill the amount details to get deposited on their account. Print details option will verify the user's money was deposited.

2.5 WITHDRAW

Real time bank machines like ATM's are mainly for withdraw of amount by the user from their account. This application provides the same feature of providing the withdraw option but it's for application-based purpose. The user will be asked to provide details of their account and print the withdraw amount from their account. If the withdraw amount is more than their balance amount in their account, the application will message an error of saying more than their balance, if not then the withdraw done message be visible to user.

2.6 RESET

This process is similar to deletion of the previous account either to create a new one or to not have any other. The Banking system applications provide user the option to erase their account details. This option will affect all other options, where user can't see the process of reset from any other option. The users name, account number, pin will be removed completely. The user can create new account if needed by create account option.

2.7 MODIFY DETAILS

ATM machines provide user to change pin, similarly user of this application can also change the pin of their account. This is called to be dynamic pin, which can be changed as of user need. Old pin verification, will lead to a page of new pin creation, where user can create a new pin. User can check the pin modification process in Print Details of Account option. User can change the pin in any number of times, there is no limit in the process of pin modification.

3. WORKING PRICIPLE AND USES

3.1 WORKING PRINCIPLE

- Simple yet complex management system made using emu8086.
- We are using emu8086 to emulate this project.
- We will create a menu bar as shown above and give the numbering.
- According to the number the user gives the emulator runs and produce the result.

3.2 USES

- User can either create or remove or update an account.
- User can deposit or withdraw money using this.
- User can get their account details by using their dynamic pin.

4. SOURCE CODE

```
;Project Title: Bank Managment System
;SOME FEATURE HIGHLIGHTS
;1.  Create Account
;2.  Deposit Money
;3.  Widthdraw Money
;4.  Print Account Details
;5.  Modify Account
;6.  Reset Account
;7.  Dynamic Pin Range
;8.  Pin Verification
;9.  Checks if account is created before performing functions
;10. When withdrawing, checks whether if u have enough money in
account

.model small
.stack 100h
.data

   dmsg1 db '    ____                __     _____              __$'
   dmsg2 db '   / __ ) ____ _ ____   / /__   / ___/ __  __ _____ / /_ ___
____ ___$'
   dmsg3 db '  / __  |/ __ `// __ \ / //_/   \__ \ / / / // ___// __// _ \ / __ `__ \$'
   dmsg4 db ' / /_/ // /_/ // / / // ,<     ___/ // /_/ /(__  )/ /_ /  __// / / / / /$'
   dmsg5 db '/_____/ \__,_//_/ /_//_/|_|   /____/ \__, //____/ \__/ \___//_/
/_/ /_/$'
   dmsg6 db '                              /____/$'

   op1mmsg1 db '   _______                 __$'
   op1mmsg2 db '  / ____/_____ ___   _____ _ / /_ ___$'
   op1mmsg3 db ' / /    / ___// _ \ / __ `// __// _ \$'
   op1mmsg4 db '/ /___ / /   /  __// /_/ // /_ /  __/$'
   op1mmsg5 db '\____//_/    \___/ \__,_/ \__/ \___/$'
```

```
op2mmsg1 db  '  ____          __           _  __$'
op2mmsg2 db '  / __ \ ___    / /_ _____ _ (_)/ /_____$'
op2mmsg3 db ' / / / // _ \ / __// __ `// // // ___/$'
op2mmsg4 db '/ /_/ //  __// /_ / /_/ // // /(__  )$'
op2mmsg5 db '/_____/ \___/ \__/ \__,_//_//_//____/$'

op3mmsg1 db ' _       __ _  __   __         __$'
op3mmsg2 db '| |     / /(_)/ /_ / /_   ____/ /_____ ____ _ _       __$'
op3mmsg3 db '| | /| / // // __// __ \ / __  // ___// __ `/| | /| / /$'
op3mmsg4 db '| |/ |/ // // /_ / / / // /_/ // /    / /_/ / | |/ |/ /$'
op3mmsg5 db '|__/|__//_/ \__//_/ /_/ \__,_//_/     \__,_/  |__/|__/$'

op4mmsg1 db '    ____                          _  __ $'
op4mmsg2 db '   / __ \ ___   ____   ____   _____ (_)/ /_$'
op4mmsg3 db '  / / / // _ \ / __ \ / __ \ / ___// // __/$'
op4mmsg4 db ' / /_/ //  __// /_/ // /_/ /(__  )/ // /_$'
op4mmsg5 db '/_____/ \___// .___/ \____//____//_/ \__/$'
op4mmsg6 db '           /_/$'

op5mmsg1 db '    __  ___           __ _  ____$'
op5mmsg2 db '   /  |/  /____   ____/ /(_)/ __/__  __$'
op5mmsg3 db '  / /|_/ // __ \ / __  // // /_ / / / /$'
op5mmsg4 db ' / /  / // /_/ // /_/ // // __// /_/ /$'
op5mmsg5 db '/_/  /_/ \____/ \__,_//_//_/   \__, /$'
op5mmsg6 db '                                /____/$'

op0mmsg1 db '   ____   __   __U _____ u$'
op0mmsg2 db 'U | __")u \ \ / /\| ___"|/$'
op0mmsg3 db ' \|  _ \/  \ V /  |  _|"$'
op0mmsg4 db '  | |_) | U_|"|_u | |___$'
op0mmsg5 db '  |____/    |_|   |_____|$'
op0mmsg6 db ' _|| \\_.-,//|(_  <<   >>$'
op0mmsg7 db '(__) (__)\_) (__)(__) (__)$'

opmsg1 db '1. Create new Account$'
opmsg2 db '2. Print Account Details$'
opmsg3 db '3. Withdraw Money $'
```

```
opmsg4 db '4. Deposit Money $'
opmsg5 db '5. Reset Account $'
opmsg6 db '6. Modify Account Details$'

opmsg8 db 'Press Enter To Return to Main Menu $'

imsg db 'What Do You Want To Do ? : $'
inputCode db ?

;Account details

accountName db 100 dup('$')
accountPIN db 100 dup('$')
accountPINcount dw 0       ;This keeps track how many digit a pin is
totalAmount dw 0
inputAmountOption db ?

;Option 1 (Create Account) Messages
op1msg1 db '1. Enter Account Name: $'
op1msg2 db '2. Enter Account Pin: $'
op1msg3 db 'Successfully Created New Account ! $'

;Option 2 <Print details> Messages
op2msg1 db 'Account Name: $'
op2msg2 db 'Currently Saved Account PIN: $'
op2msg3 db 'No Accounts Currently Saved !$'
op2msg4 db 'Total Money Left: $'
op2msg5 db 'You Have No Money $'

;Option 4 <Money> Messages
op4msg1 db '1. Rs 1000$'
op4msg2 db '2. Rs 2000$'
op4msg3 db '3. Rs 5000$'
op4msg4 db '4. Rs 10000$'
op4msg5 db 'Enter Code: $'
op4msg6 db 'You Are Withdrawing Too MUCH !$'
```

```
    ;Option 5 <Reset> Messages
    op5msg1 db 'Account Has been reset successfully$'

    ;Option 6 <Modify Account> Messages
    op6msg0   db 'Account Details Successfully Changed !$'
    op6msg1_1 db '1. New Account Name ( old: $'
    op6msg1_2 db ' ) : $'
    op6msg2_1 db '2. New Account Pin ( old: $'
    op6msg2_2 db ' ) : $'

    ;PIN Protection
    pinop_msg1 db 'Enter PIN: $'
    pinop_msg2 db 'Account NOT created ... $'

.code

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                        ;
;                       U T I L S                        ;
;                                                        ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

;Enter to Continue
proc etc
   etcin:
      mov ah,1
      int 21h
      cmp al,13
      je mainloop
      jmp etcin
   ret
etc endp

;This checks whether the account has been created or not using the pin
Count
checkAccountCreated proc
  cmp accountPINcount,0
```

```
  je accountNotCreated
  ret

  accountNotCreated:
    call clearScreen
    printString pinop_msg2
    call etc

checkAccountCreated endp

;just mov number to ax and call this proc
printNumber PROC
    ;initilize count
    mov cx,0
    mov dx,0
    label1:
        ; if ax is zero
        cmp ax,0
        je print1

        ;initilize bx to 10
        mov bx,10

        ; extract the last digit
        div bx

        ;push it in the stack
        push dx

        ;increment the count
        inc cx

        ;set dx to 0
        xor dx,dx
        jmp label1

    print1:
```

```
        ;check if count
        ;is greater than zero
        cmp cx,0
        je exitprint

        ;pop the top of stack
        pop dx

        ;add 48 so that it
        ;represents the ASCII
        ;value of digits
        add dx,48

        ;interuppt to print a
        ;character
        mov ah,02h
        int 21h

        ;decrease the count
        dec cx
        jmp print1
exitprint:
ret
printNumber ENDP

clearScreen proc near
   call newLine
   call newLine
   ret
clearScreen endp

newLine proc near
   mov ah,2
   mov dl,10
   int 21h
   mov dl,13
   int 21h
```

```
    ret

newLine endp

macro printString str
  mov ah,9
  lea dx,str
  int 21h
endm

;Ask for user pin here
getPinInput proc
  call clearScreen

  printString pinop_msg1

  mov si,offset accountPIN
  mov cx,accountPINcount     ;Search n amount of times the pin Count
  getinput:

    mov ah,7
    int 21h

    cmp al,[si]

    mov dl,'*'
    mov ah,2
    int 21h

    jne mainloop

    inc si
  loop getinput

  ret
getPinInput endp
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                            ;
;                   M E N U   SYSTEM                         ;
;                                                            ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

DisplayMenu proc near
    printString dmsg1
    call newLine
    printString dmsg2
    call newLine
    printString dmsg3
    call newLine
    printString dmsg4
    call newLine
    printString dmsg5
    call newLine
    printString dmsg6
    call newLine

    call newLine
    printString opmsg1
    call newLine
    printString opmsg2
    call newLine
    printString opmsg3
    call newLine
    printString opmsg4
    call newLine
    printString opmsg5
    call newLine
    printString opmsg6
    call newLine
    ret
DisplayMenu endp
```

```
GetInputMenuSystem proc near
   call newLine
   printString imsg
   mov ah,1
   int 21h
   mov inputCode,al
   ret
GetInputMenuSystem endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                   ;
;           O P T I O N  1   => CREATE ACCOUNT              ;
;                                                   ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

macro ISop11 str
   mov si,offset str
   input:
      mov ah,1
      int 21h
      cmp al,13
      je labelop1_1
      mov [si],al
      inc si
      jmp input

   exitMac:
      ret

endm

macro ISop12 str
   mov si,offset str
   input2:
      mov ah,1
      int 21h
      cmp al,13
```

```
        je labelop1_2
        inc accountPINcount
        mov [si],al
        inc si
        jmp input2

    exitMac2:
        ret

endm

proc etcop1
   etcop1in:
      mov ah,1
      int 21h
      cmp al,13
      je mainloop
      jmp etcop1in
   ret
etcop1 endp

op1 proc

    call clearScreen

    printString op1mmsg1
    call newLine
    printString op1mmsg2
    call newLine
    printString op1mmsg3
    call newLine
    printString op1mmsg4
    call newLine
    printString op1mmsg5
    call newLine
    call newLine
    call newLine
```

```
    printString op1msg1
    ISop11 accountName

    labelop1_1:
      call newLine
      printString op1msg2
      ISop12 accountPIN

    labelop1_2:

        call newLine
        call newLine
        printString op1msg3
        call etcop1

    ret
op1 endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                              ;
;         O P T I O N  1   => PRINT DETAILS              ;
;                                                              ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

proc etcop2
   call newLine
   printString opmsg8
   etcop2in:
     mov ah,1
     int 21h
     cmp al,13
     je mainloop
     jmp etcop2in
   ret
etcop2 endp
```

```
op2 proc

  call checkAccountCreated  ;check whether the account has been created
or not
  call getPinInput  ;gets the pin input for verification
  call clearScreen

  printString op2mmsg1
  call newLine
  printString op2mmsg2
  call newLine
  printString op2mmsg3
  call newLine
  printString op2mmsg4
  call newLine
  printString op2mmsg5
  call newLine
  call newLine
  call newLine

  printString op2msg1
  printString accountName
  call newLine

  printString op2msg2
  printString accountPIN
  call newLine

  printString op2msg4
  mov ax,totalAmount
  cmp ax,0
  je noMoneyError
  call printNumber
  call newLine

  call etcop2
```

```
  noMoneyError:
    printString op2msg5
    call newLine
    call etcop2

  ret

op2 endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                            ;
;              O P T I O N  3   => WIDTHDRAW MONEY            ;
;                                                            ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

op3 proc

  call checkAccountCreated  ;check whether the account has been created
or not
  call getPinInput  ;gets the pin input
  call clearScreen


  printString op3mmsg1
  call newLine
  printString op3mmsg2
  call newLine
  printString op3mmsg3
  call newLine
  printString op3mmsg4
  call newLine
  printString op3mmsg5
  call newLine
  call newLine
  call newLine

  printString op4msg1
```

```
call newLine
printString op4msg2
call newLine
printString op4msg3
call newLine
printString op4msg4
call newLine

call inputAmountCode

cmp inputAmountOption,'1'
je wcop1

cmp inputAmountOption,'2'
je wcop2

cmp inputAmountOption,'3'
je wcop3

cmp inputAmountOption,'4'
je wcop4

;check if widraw amount <= totalAmount in acc
wcop1:
  mov bx,totalAmount
  cmp bx,1000
  jl nowaybro
  sub totalAmount,1000
  jmp mainloop
wcop2:
  mov bx,totalAmount
  cmp bx,2000
  jl nowaybro
  sub totalAmount,2000
  jmp mainloop
wcop3:
  mov bx,totalAmount
```

```
    cmp bx,5000
    jl nowaybro
    sub totalAmount,5000
    jmp mainloop
  wcop4:
    mov bx,totalAmount
    cmp bx,10000
    jl nowaybro
    sub totalAmount,10000
    jmp mainloop

  ;error message for widthdrawing too much
  nowaybro:

    call newLine
    call newLine
    printString op4msg6
    call etcop4

  ret

op3 endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                            ;
;           O P T I O N  4   => DEPOSIT MONEY                    ;
;                                                                            ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

proc etcop4
   call newLine
   ;printString opmsg8
   etcop4in:
      mov ah,1
      int 21h
      cmp al,13
      je mainloop
```

```
    jmp etcop4in
  ret
etcop4 endp

proc inputAmountCode
  call newLine
  printString op4msg5
  mov ah,1
  int 21h
  mov inputAmountOption,al
  ret
inputAmountCode endp

op4 proc

  call checkAccountCreated  ;check whether the account has been created
or not
  call getPinInput  ;gets the pin input
  call clearScreen

  printString op4mmsg1
  call newLine
  printString op4mmsg2
  call newLine
  printString op4mmsg3
  call newLine
  printString op4mmsg4
  call newLine
  printString op4mmsg5
  call newLine
  printString op4mmsg6
  call newLine
  call newLine
  call newLine

  printString op4msg1
  call newLine
```

```
printString op4msg2
call newLine
printString op4msg3
call newLine
printString op4msg4
call newLine

call inputAmountCode

cmp inputAmountOption,'1'
je dcop1

cmp inputAmountOption,'2'
je dcop2

cmp inputAmountOption,'3'
je dcop3

cmp inputAmountOption,'4'
je dcop4

dcop1:
  add totalAmount,1000
  jmp mainloop
dcop2:
  add totalAmount,2000
  jmp mainloop
dcop3:
  add totalAmount,5000
  jmp mainloop
dcop4:
  add totalAmount,10000
  jmp mainloop

ret

op4 endp
```

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                     ;
;            O P T I O N  5   => RESET ACCOUNT                ;
;                                                     ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

proc etcop5
   call newLine
   ;printString opmsg8
   etcop5in:
      mov ah,1
      int 21h
      cmp al,13
      je mainloop
      jmp etcop5in
   ret
etcop5 endp

op5 proc

  call checkAccountCreated  ;check whether the account has been created
or not
  call getPinInput  ;gets the pin input

  ;Do the rest of the work .. display the data
  call clearScreen

  mov si,offset accountName
  mov cx,30
  l1:
    mov [si],' '
    inc si
  loop l1

  mov cx,30
  mov si,offset accountPIN
```

```
  l2:
    mov [si],' '
    inc si
  loop l2

  mov totalAmount,0
  mov accountPINcount,0 ;reset pin count

  printString op5msg1
  call etcop5
  ret
op5 endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                          ;
;            O P T I O N  6   => MODIFY ACCOUNT DETAILS         ;
;                                                                          ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;

proc etcop6
   call newLine
   ;printString opmsg8
   etcop6in:
      mov ah,1
      int 21h
      cmp al,13
      je mainloop
      jmp etcop6in
   ret
etcop6 endp

macro ISop6 str
 mov si,offset str
    ISop6input:
        mov ah,1
        int 21h
        cmp al,13
```

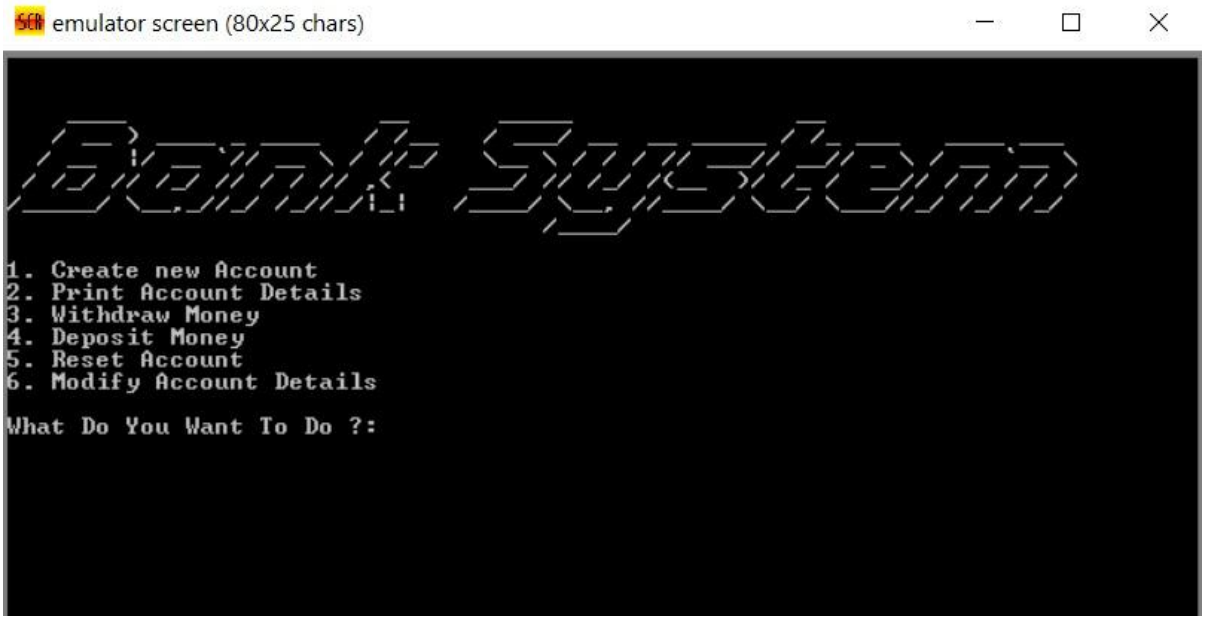
```
        je labelop6_1
        mov [si],al
        inc si
        jmp ISop6input
endm

macro ISop6_2 str
 mov si,offset str
 mov accountPINcount,0 ;reset pin count
    ISop6_2input:
        mov ah,1
        int 21h
        cmp al,13
        je labelop6_2
        inc accountPINcount ;increment pin account again
        mov [si],al
        inc si
        jmp ISop6_2input
endm

op6 proc

  call checkAccountCreated  ;check whether the account has been created
or not
  call getPinInput  ;gets the pin
  call clearScreen

  printString op5mmsg1
  call newLine
  printString op5mmsg2
  call newLine
  printString op5mmsg3
  call newLine
  printString op5mmsg4
  call newLine
  printString op5mmsg5
  call newLine
```

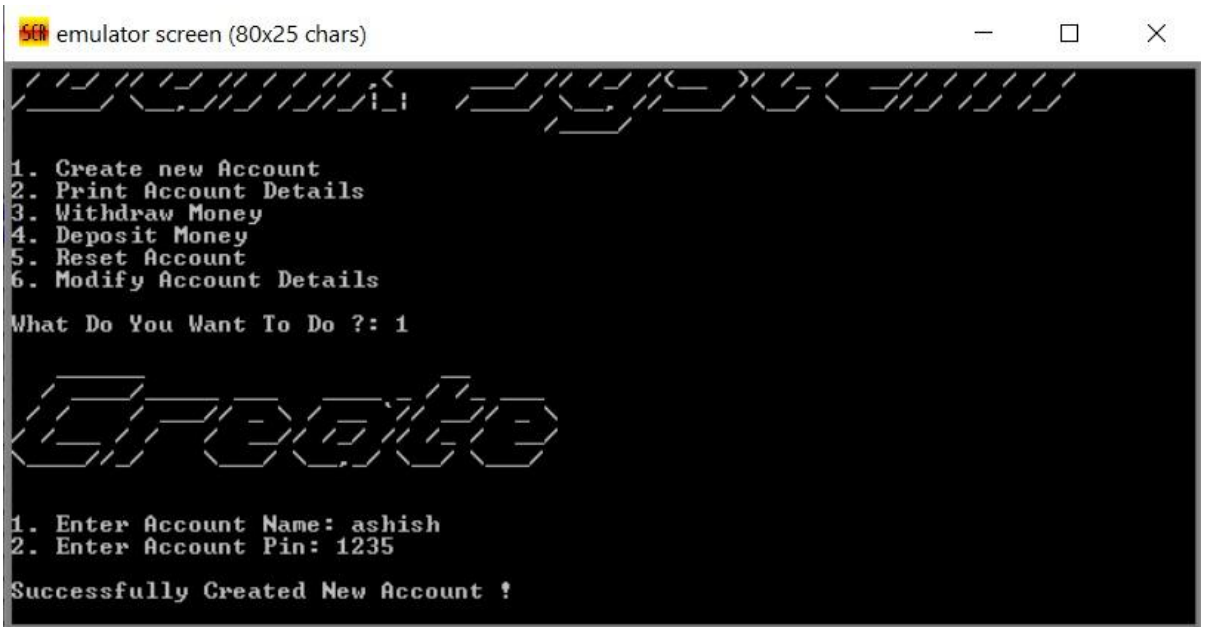
```
 printString op5mmsg6
 call newLine
 call newLine
 call newLine

 ;;account name
 printString op6msg1_1
 printString accountName
 printString op6msg1_2

 ISop6 accountName ;input accountName

 labelop6_1:

   call newLine
   printString op6msg2_1
   printString accountPIN
   printString op6msg2_2
   ISop6_2 accountPIN

 labelop6_2:

   ;Finished MSG
   call newLine
   call newLine
   printString op6msg0
   call etcop6


 ret
op6 endp

;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;                                                                  ;
;                 E N T R Y   P O I N T                            ;
;                                                                  ;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
```

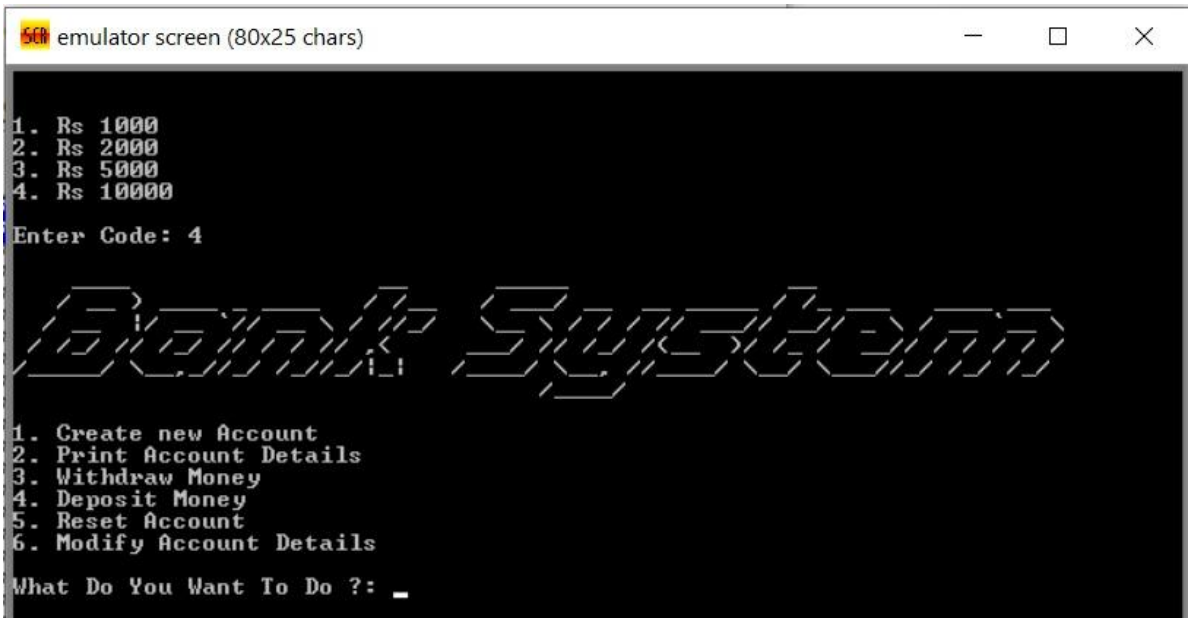
```
Main proc

    mov ax,@data
    mov ds,ax

    mainloop:

        call clearScreen
        call DisplayMenu
        call GetInputMenuSystem

        cmp inputCode,'0'
        je exit

        cmp inputCode,'2'
        je op2

        cmp inputCode,'4'
        je op4

        cmp inputCode,'3'
        je op3

        cmp inputCode,'6'
        je op6

        cmp inputCode,'1'
        je op1

        cmp inputCode,'5'
        je op5

        jmp mainloop

    exit:
```

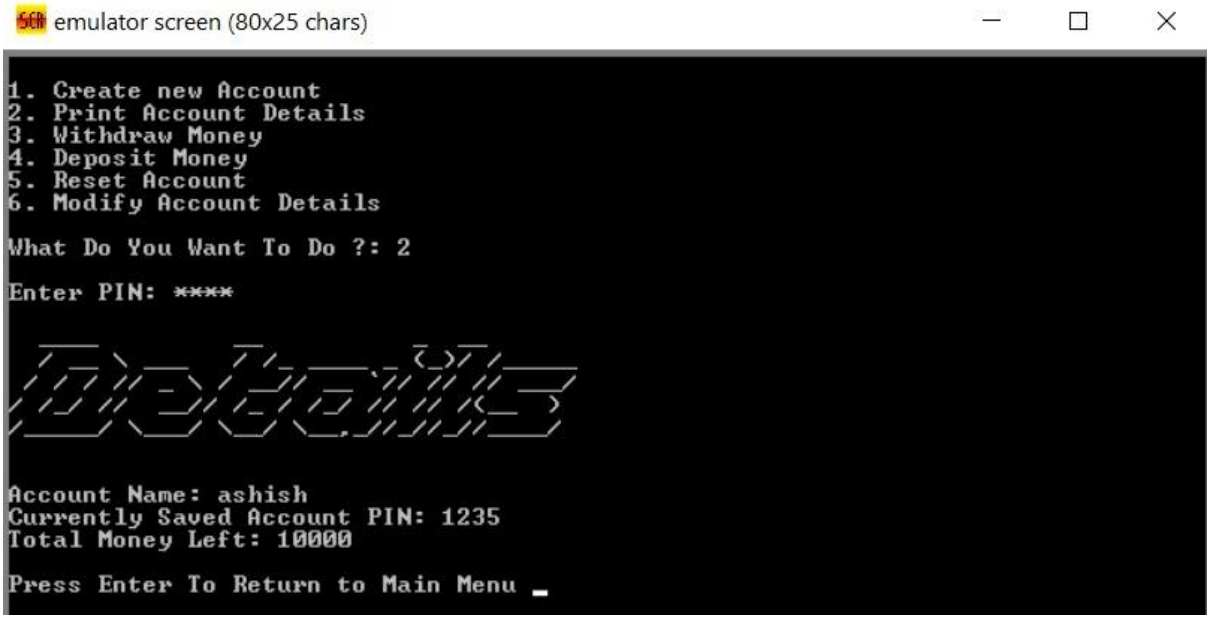
```
    call newLine
    call newLine

    printString op0mmsg1
    call newLine
    printString op0mmsg2
    call newLine
    printString op0mmsg3
    call newLine
    printString op0mmsg4
    call newLine
    printString op0mmsg5
    call newLine
    printString op0mmsg6
    call newLine
    printString op0mmsg7
    call newLine

    call newLine

    mov ah,4ch
    int 21h

  main endp
end main
```

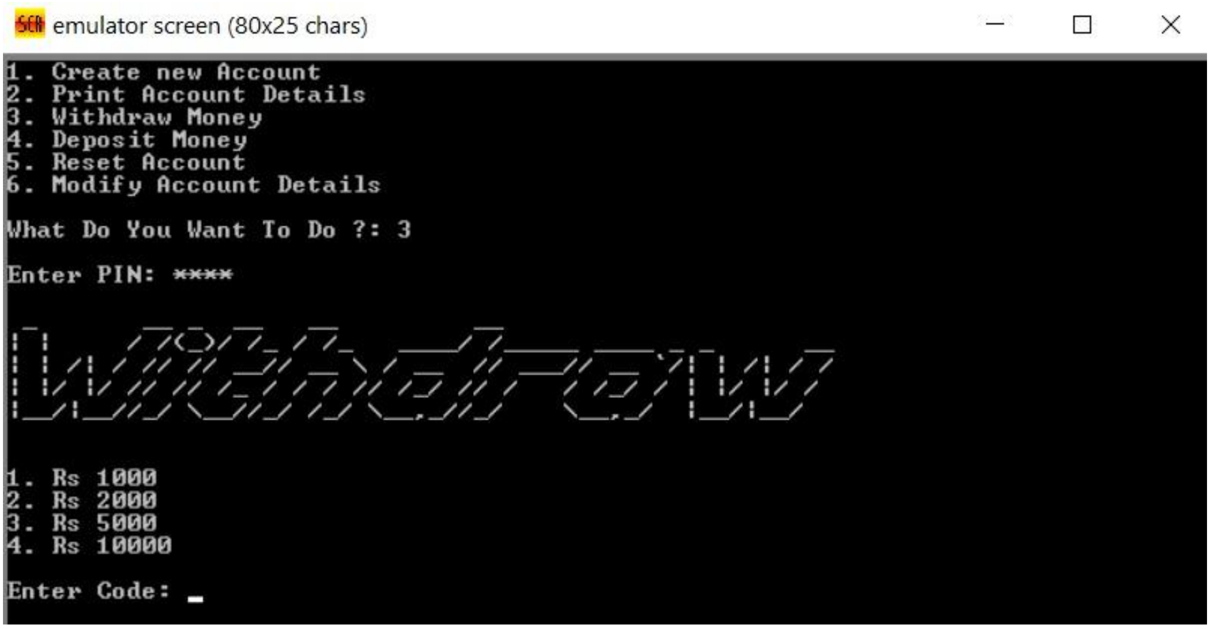
5. SCREENSHOTS

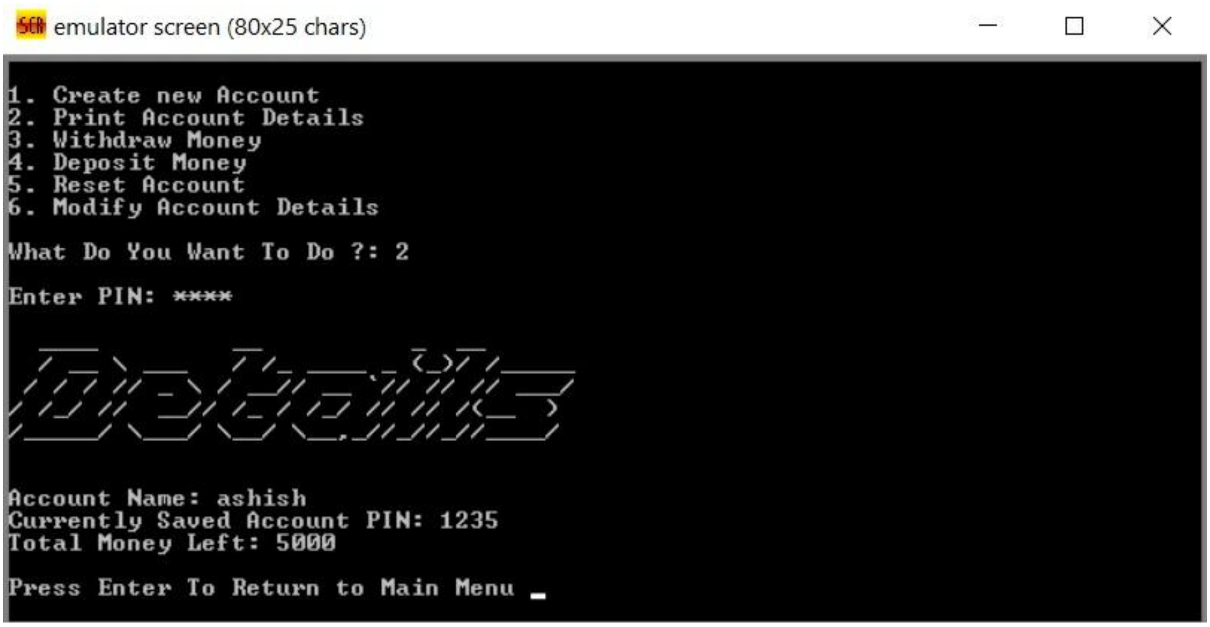




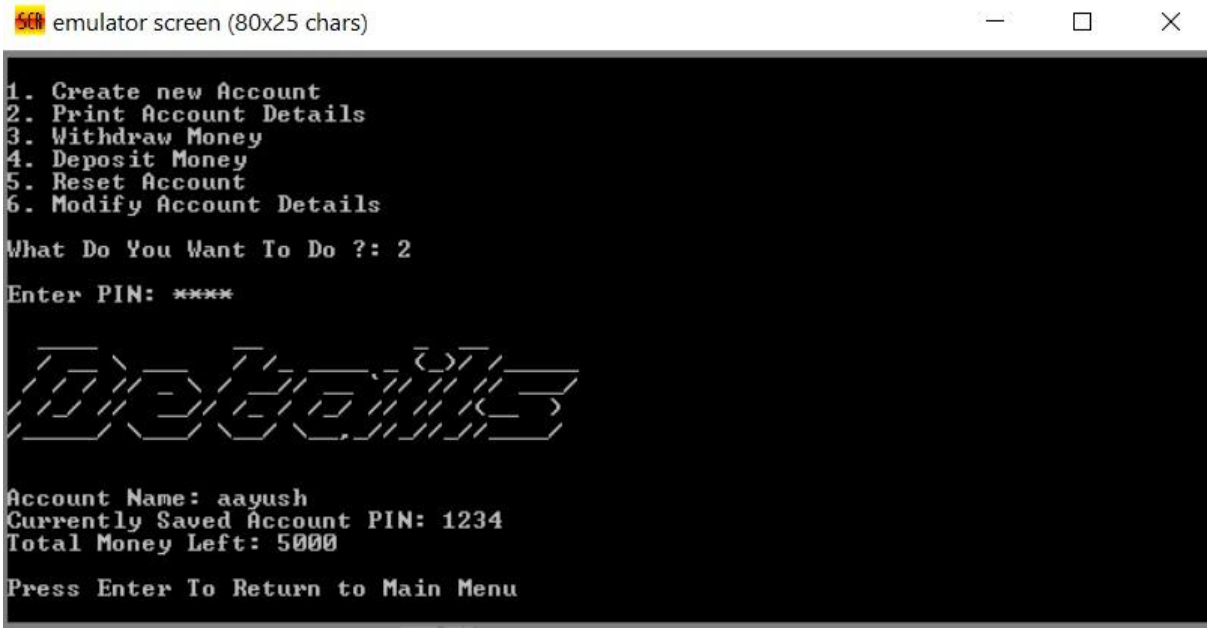

emulator screen (80x25 chars)
1. Rs 1000
2. Rs 2000
3. Rs 5000
4. Rs 10000
Enter Code: 4
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?:

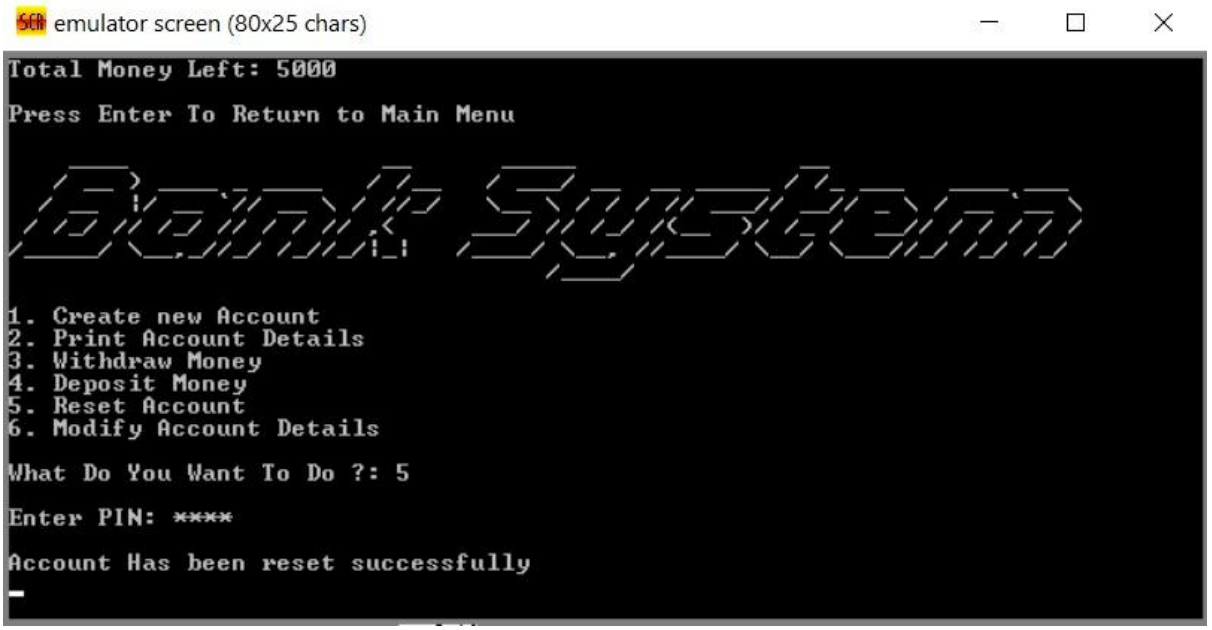

emulator screen (80x25 chars)
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?: 2
Enter PIN: ****
Account Name: ashish
Currently Saved Account PIN: 1235
Total Money Left: 10000
Press Enter To Return to Main Menu

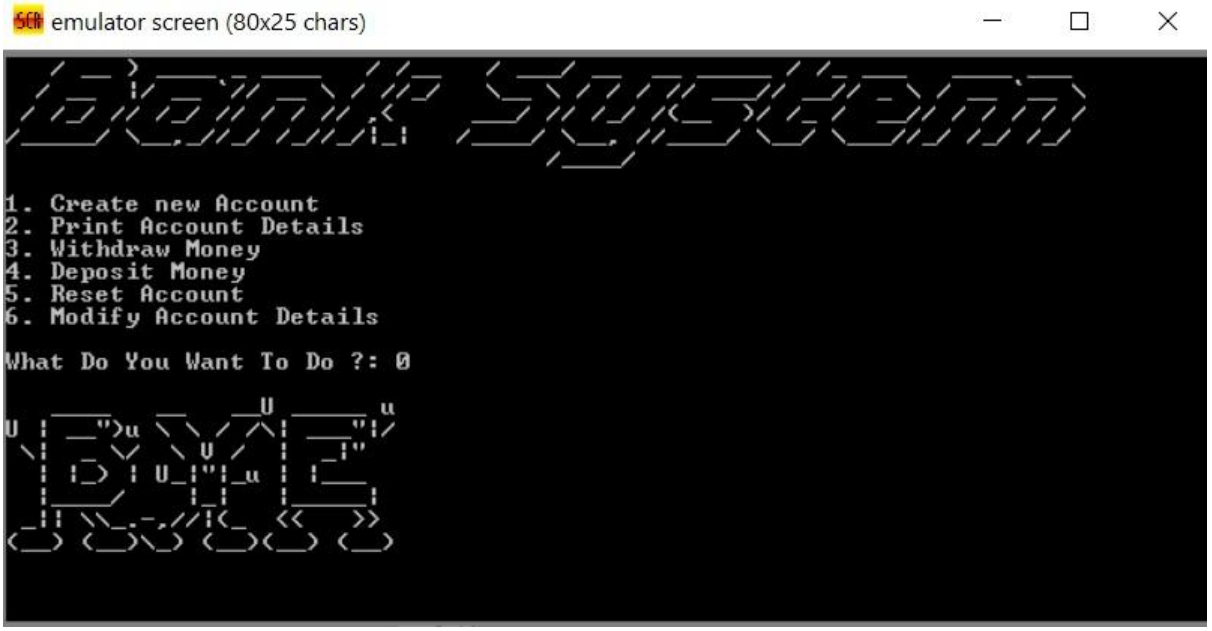

emulator screen (80x25 chars)
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?: 3
Enter PIN: ****
1. Rs 1000
2. Rs 2000
3. Rs 5000
4. Rs 10000
Enter Code:


emulator screen (80x25 chars)
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?: 2
Enter PIN: ****
Account Name: ashish
Currently Saved Account PIN: 1235
Total Money Left: 5000
Press Enter To Return to Main Menu


emulator screen (80x25 chars)
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?: 6
Enter PIN: ****
1. New Account Name (old: ashish): aayush
2. New Account Pin (old: 1235): 1234


emulator screen (80x25 chars)
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?: 2
Enter PIN: ****
Account Name: aayush
Currently Saved Account PIN: 1234
Total Money Left: 5000
Press Enter To Return to Main Menu


emulator screen (80x25 chars)
Total Money Left: 5000
Press Enter To Return to Main Menu
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?: 5
Enter PIN: ****
Account Has been reset successfully


emulator screen (80x25 chars)
1. Create new Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details
What Do You Want To Do ?: 0

6. TOOLS USED

- **Emulator:-** An emulator typically enables the host system to run software or use peripheral devices designed for the guest system. Emulation refers to the ability of a computer program in an electronic device to emulate another program or device. The Church-Turing thesis implies that any operating environment can be emulated within any other environment.

  Benefits of using an emulator

1. Better graphics than the original hardware.
2. Emulators maintain the original look and behavior of the digital data.
3. Emulators are cost effective over time.
4. Reduces labor hours, because rather than continuing an ongoing task of continual data migration for every digital object, once the library of past and present operating systems and application software is established in an emulator, these same technologies are used for every document using those platforms.
5. Many emulators have already been developed and released under the GNU General Public License through the open source environment, allowing for wide scale collaboration.

- **Assembly language:-** An **assembly language** is a low-level programming **language** designed for a specific type of processor. It may be produced by compiling source **code** from a high-level programming **language** (such as C/C++) but can also be written from scratch.

7. LITERATURE REVIEW

- A study about the e-banking over 1999–2006 shows that the application of e-banking can improve banks' performance in terms of the growth in assets, reduction in operating expenses and portfolio enhancement.

- Even in 1990s, Sraeel (1996) emphasises that creating virtual banking will not only create a new service delivery channel, but also lead to value creation to both banks and customers.

- Through interviewing banks in a small island and examining their e-banking websites from 2004 to 2006, Jenkins (2007) indicates that those banks were using e-banking as an assurance to their customers to maintain a competitive quality of service.

- **V. Raja, Joe A. (2012), "Global e-banking scenario and challenges in banking system",**
  This paper is an attempt to explore the various levels of internet banking services provided by banks using the secondary data. It also compares the traditional banking systems with net banking. It lists out the various advantages of internet banking and the successful security measures adopted by different banks for secured banking transactions. It also analyzes how E-banking can be useful for banking industry during this global financial melt down.

- Lu et al. (2005) reveal that one of the key strategic responses of banks in China before joining WTO was to develop e-banking to a more competitive environment, even under the current condition of lack of practical customer credit system.

- Lu et al. (2005) reveal that one of the key strategic responses of banks in China before joining WTO was to develop e-banking to a more competitive environment, even under the current condition of lack of practical customer credit system.

- **Agarwal R., Rastogi S., Mehrotra A., (2009), "Customers' perspectives regarding ebanking in an emerging economy"**
  Determining factors affecting customer perception and attitude towards and satisfaction with e-banking is an essential part of a bank's strategy formulation process in an emerging economy like India. To gain this understanding in respect of Indian customers, the study was conducted on respondents taken from the northern part of India. The major findings depict that customers are influenced in their usage of e-banking services by the kind of account they hold, their age and profession, attach highest degree of usefulness to balance enquiry service among e-banking services, consider security & trust most important in affecting their satisfaction level and find slow transaction speed the most frequently faced problem while using e-banking.

- **Kenneth B. Y., David H. W., Claire L., Randall B, (2010) "Offline and online banking - where to draw the line when building trust in e-banking?",** found that Traditional service quality builds customer trust in the e-banking service. The size and reputation of the bank were found to provide structural assurance to the customer but not in the absence of traditional service quality. Web site features that give customers confidence are significant situation normality cues.

- Calisir and Gumussoy (2008) compare the consumer perception of internet banking and other banking channels and report that internet banking, ATM and phone banking substitute each other.

- Polatoglu and Ekin (2001) reported that, since 1997 several leading Turkish banks have offered online banking services successfully. According to the Turkey banks association, 27 out of a total of 47 banks, in other words 58% of all banks in Turkey were offering internet banking services in 2006.

- Guerrero et al. (2007) examine the usage of internet banking by Europeans and their results indicate that ownership of diverse financial products and services, attitude towards finances and trust in the internet as a banking channel influence clients' usage of internet banking.