VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

AUTOMATIC BANK MANAGEMENT SYSTEM

BY:

N. ANAND VENKATA SUBBARAJU (19BCE0264)

N.L. NARASIMHA RAJU (19BCE2247)

KOMMINENI BHARGAV (19BCE0322)

K. PAVAN TEJA VARMA (19BCB0026)

MANAV KULSHRESTHA (19BCE0718)

PROJECT REPORT OF

CSE-2006 MICROPROCESSOR AND INTERACING

FALL SEMESTER 2021-22

Submitted To:

Faculty : Naveen Mishra **Signature :**

Date : 3/12/2021

Slot : E2

1.ABSTRACT:

The Bank Account Management System is an software for retaining a person's account in a bank.

In this project we attempted to expose the running of a banking account gadget and cowl the primary capability of a Bank Account Management System.

To expand a assignment for fixing monetary programs of a patron in banking surroundings for you to nurture the desires of an give up banking consumer through supplying diverse methods to carry out banking tasks.

Also to permit the consumer's paintings area to have extra functionalities which aren't supplied beneath a traditional banking assignment.

The fundamental intention of this project is to expand software program for Bank Account Management System.

This project has been evolved to perform the approaches effortlessly and quickly, which isn't feasible with the manuals structures, that are conquer through this software program.

This project is evolved the use of 8086 meeting language the use of emu 8086. Creating and handling necessities is a assignment of IT, structures and product improvement initiatives or certainly for any hobby wherein you need to manipulate a contractual relationship.

Organization want to efficaciously outline and manipulate necessities to make sure they're assembly desires of the patron, at the same time as proving compliance and staying at the agenda and inside budget.

The gadget is then designed according with specs to ful fill the necessities. The gadget layout is then carried out with 8086.

2. INTRODUCTION:

To develop a software 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 users workspace to have additional functionalities which are not provided under a conventional banking software.

In this project we are going to explain about Banking Management System. This project has facility to opening account, depositing and withdrawing money. The Bank management system is an application for maintaining a person's account in a bank. The system provides the access to the customer to create an account, deposit/withdraw the cash from his account, also to view reports of all accounts present. The following presentation provides the specification for the system.

LITERATURE SURVEY

TITLE	YEAR OF PUBLICATION	AUTHOR	WORK DONE	INFERENCES/ RECOMMDATIONS
An overview of Microprocessors and assembly language programming	2017	Abdullah Al Zaman, Nusrat Jahan Monira	They have focused on the evolution of the microprocessors first, and then went for the categorization, organization, operationand some other fundamental things. They also discussed about the several cycles thata microprocessor goes through and at last gave some	1.The programmer requires knowledge of the processor architecture and instruction set. 2.Machine language coding is difficult to understand. 3. Many instructions are required

			ideas and aspects of assembly language programming.	to achieve small tasks. 4.Source programs tend to be large and difficult to follow.
User Experience in using MOOC of Assembly Language Programming	2019	Nuridawati Md Ezal, Svzilwati Mohammad, Nuraiza Ismail	This paper covers the topics on eneral-purpose microprocessor, its rchitecture and system organization, Through the MOOC (Massive Open Online Course) platform.	This paper comprises a quantitative and qualitative exploration of Microprocesso r students' views on the use of MOOC in Microprocesso r course learning activity. The specific MOOC issues investigated included: existing usage; student's perception on the MOOC design and content; and the impact of MOOC on student's learning activity.

Analysis of the development of smart banking in the banking industry in recent years	2021	KELVIN CHENG	This paper mainly analyzes the factors causing online banking development and theeconomic impacts in the banking industry based on the statistical data and relevant research study, to evaluate its implications today's economic and business situations.	Online banking remains to demonstrate a strong expansion trend with 270% growth of market size in 5 years, it has become more valuable and influential to various stakeholders. The Covid-19 pandemic leads to a shift in customer behaviour from traditional to digital style to fulfiltheir needs, traditional businessfirms are forced to develop theirown digital business model for entering the banking industry.

Study on the course-Assembly Language Programming	2013	LI LIU	It mainly focus on core course with strong theoretical and practical features. It explains the basic framework and the working procedure of the micro computer, discusses the assembly language programming technologies to the students. And it is a leader course for manyother professional courses.	1. Design of evaluation mode 2. Deepening and broadening of the teaching content 3.A variety of assembly language development tools
Teaching of 8088/86 Programming with 8086 Assembly Emulator	2020	San H laingo Khin Trar Trar Soe	The course objective is "to develop the applicable programs for interfacing input/output devices with the target microprocessor" and course learning outcomes are to	1. 8088/86 are 16-bit microprocessors, which have 1M byte of memory. 2. Before programming concepts have to be learnt, internal configurations of 8088/86

			describe architectures and features of microprocessors, to illustrate programming proficiency with assembly language and to demonstrate interfacing with input/output devices.	microprocesso r, especially internal registers withtheir functions, operation capabilities and limitations must be known
Teaching Research on Assembly Language Course Based on Applied Talents Training	2017	Mengquin Feng, Lijuan Quin	Assembly language plays an important role in the training of computer application talents. The paper focuses on the application of teaching talents and the aspects of ALP	Challenges, such as obsolescence, backward teaching methods, unclear curriculum goals, and so on. Therefore, the need for curriculum reform to make it meet the needs of computer application-oriented personnel training.

K Semantics for Assembly Languages: A Case Study	2014	Mihail Asăvoae	n this paper, they used the K framework to formally define a MIPS-based assembly language. Their proposed definition is modular in the sense that it accommodates various organizations for the storage related aspects of the semantics. They also present how to instantiate our K language definition on two main memory models, corresponding to different representations of the assembly code. Such a formal language definition could be directly used by the program verification tools.	The main memory modeling is more complex than what is presented. Only a limited amount of automated testing is allowed, subjective to certain assumptions. This presents a limited subset of the x86 assembly language for malware behavior detection

An 8-bit Scientific Calculator Based Intel 8086 Virtual Machine Emulator	2019	Qasem Abu Al Haija, Saleh Al-Abdulatif and Mohaned Al-Ghofaily	In this paper, they propose an eight bit scientific calculator based on Intel 8086 assembly language programming. The calculator was designed over the virtual machine for Intel 8086 icro sor using EMU8086 emulator software. Several arithmetic and logic operations as well as trigonometric functions were implemented in this paper.	The work in this paper can be improved by implementing more the integration of the functions and add the function plot tool which are under-consideration and extending the capabilities of thecalculator to allow a 16-bit calculations as well as add more arithmetic operations.

3.OBJECTIVE:

Although the basic type of services offered by a bank depends upon the type of bank and the country, services provided usually include: Taking deposits from their customers and issuing current or checking accounts and savings accounts to individuals and business. There may be many human errors during this process. So, we will develop a computerized system based on this simple bank processes.

4.PROBLEM STATEMENT:

To develop a system that will overlook the activities going transaction the particular bank without manual processing. All transaction will be updated automatically by using the information stored in record. The main motive behind this project is to develop a system which will able to handle the overall tasks going inside the institutions without much effort.

5.EXISTING PROBLEMS:

The existing system work manually. The existing system has got lot of intricacies within itself and need lot of human effort and paper works. All above the data need to be maintained on ledgers and maintaining this is a tedious and risky process. As the transaction's increases, so the data too. So, the task of maintaining them increases exponentially. To view a data may need lot of paper to be searched. Some of the negative aspects of the existing system are as follows:

1)Time Consuming:

There is a lot of time consumes in the bank, whenever we open account, deposit, withdraw or pass a loan than because of many customers with his/ her different purpose, than we wait for our turn sometimes 2 to 3 hours.

2)Reliability:

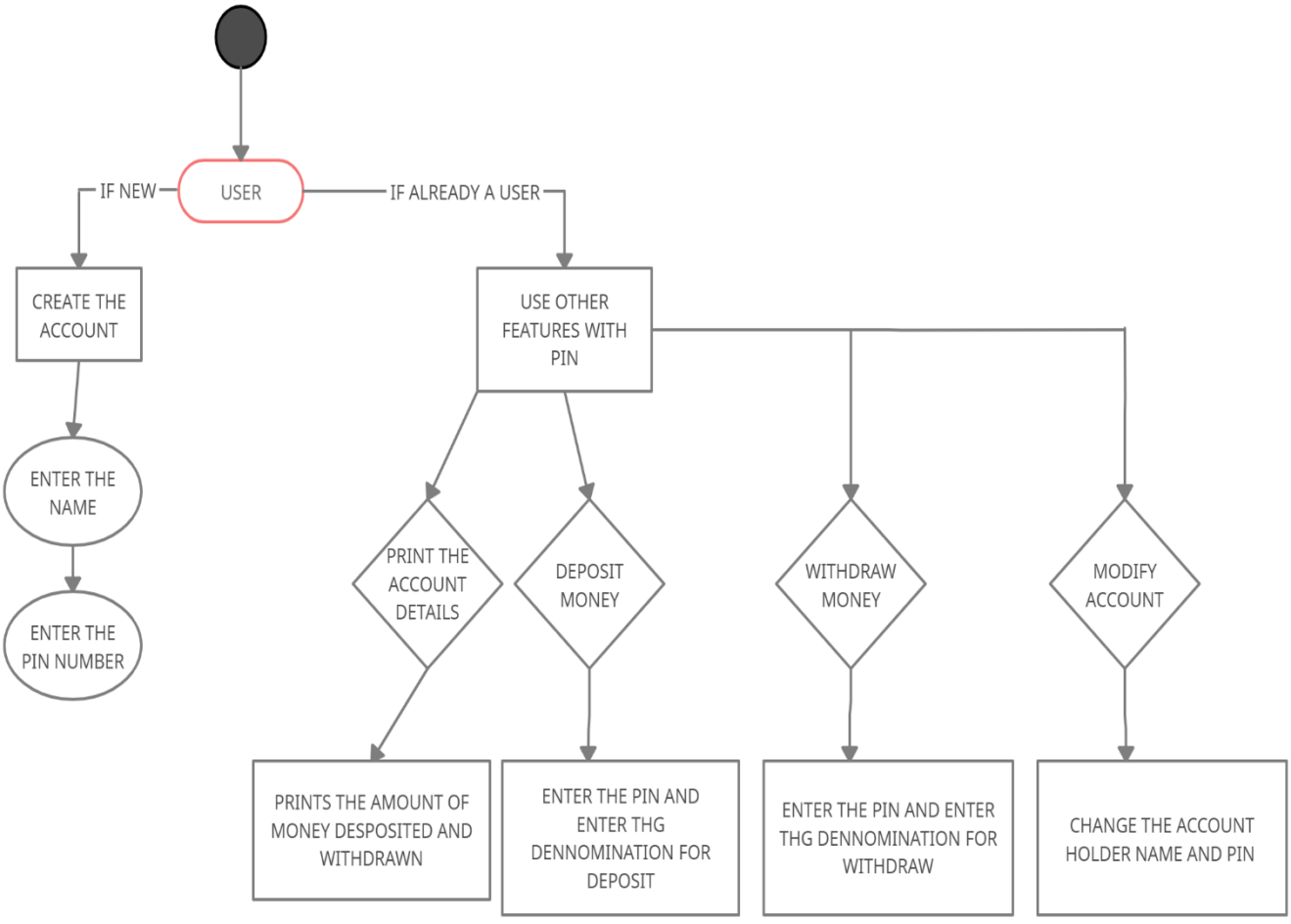
This banking system is not fully reliable whenever the computer system is creating a problem and not work properly than sometime our data is damaged or lost.

3)Less Accurate:

This system is not fully accurate, because sometime computerized system creates a problem in working, then the computer system also gives us wrong results.

To overcome these, the proposed system has been suggested.

6.BLOCK DIAGRAM:



7.ALGORITHM:

- Start the program
- The menu displays 5 options.
- First, we must create an account in the system
- Then we must enter the amount that must be deposited
- Then we can either withdraw the money or we can deposit the money.
- The amounts that can be 1000 5000 and 10000
- If
- The customer wants to change their name or the pin
- There is an option to change the pin and name.
- If
- The customer wants to reset account.
- THERE IS AN OPTION THAT HE CAN RESET THEIR ACCOUNT
- IF
- THE CUSTOMER ENTERS AN OPTION THAT IS IRRELEVANT TO THE MENU
- IT DISPLAYS THE ERROR
- IF NO
- END

8.PROGRAM CODE:

```
.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
```

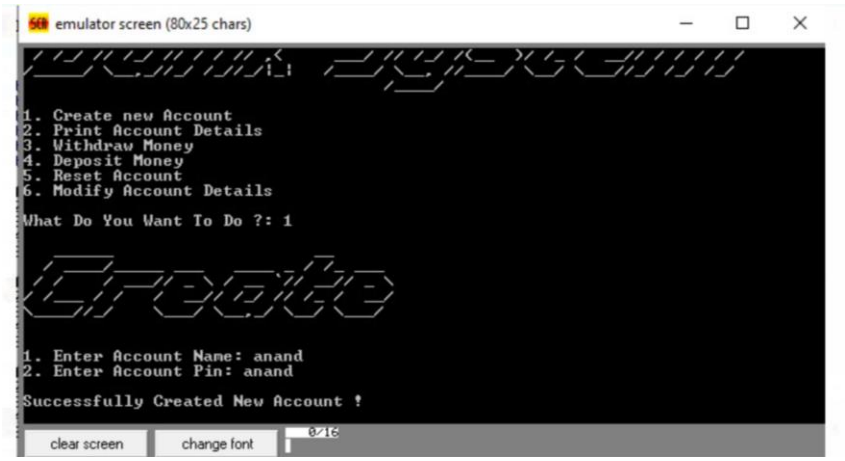
```
call checkAccountCreated  ;check whether the account has been created or not
call getPinInput  ;gets the pin
call clearScreen

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'
```

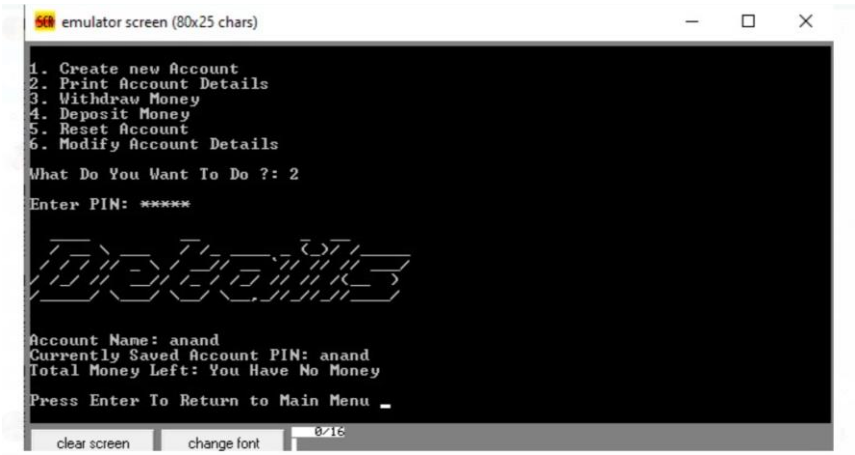
```
    je op4

    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
```

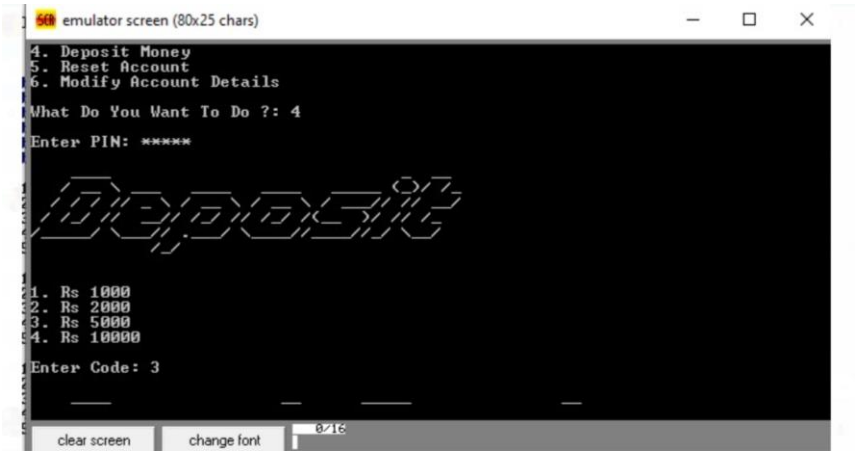
HOME PAGE TO CREATE AN ACCOUNT



ACCOUNT DETAILS

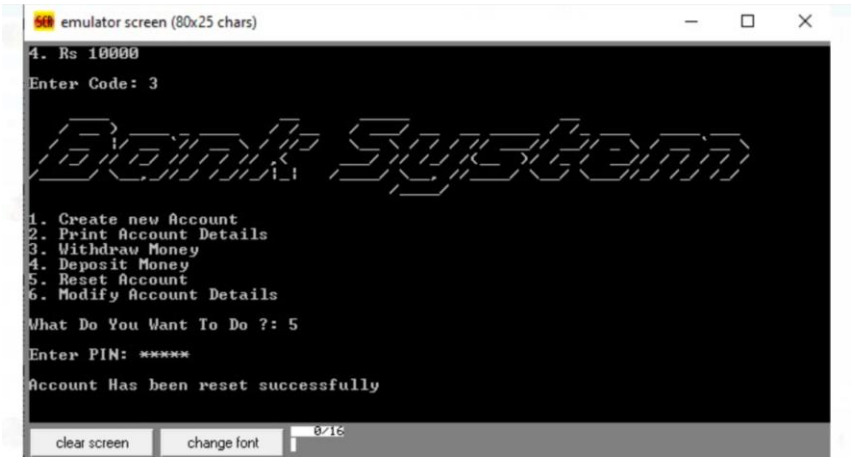


DEPOSIT DETAILS

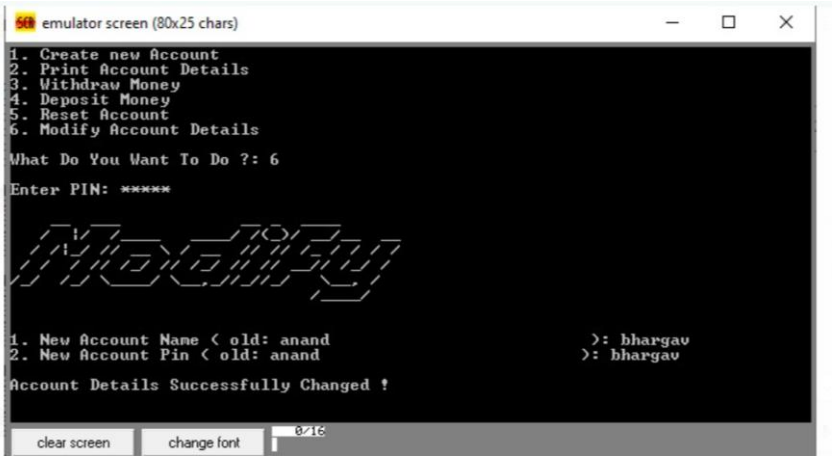


WITHDRAW DETAILS

ACCOUNT RESET:



ACCOUNT MODIFICATION:



9.DRAWBACKS:

1. Single user at a time
2. No proper robust menu system
3. Account cannot be modified once it is reset

10.CONCLUSION

- This project is developed to nurture the needs of a user in a banking sector by embedding all
- The tasks of transactions taking place in a bank.

- This project helps us for better understanding of the compiler's output.

10. REFERENCES

https://www.youtube.com/watch?v=1FXhjErUz58&ab_channel=jav idx9

https://www.youtube.com/watch?v=zEuvNYe7WG0&ab channel=R asimMuratovic

https://www.freecodecamp.org/news/what-are-assembly

languages/ https://csenotesforyou.blogspot.com/2016/12/assembly language-program-development.html

An 8-bit Scientific Calculator Based Intel 8086 Virtual Machine Emulator

K Semantics for Assembly Languages: A Case Study

Teaching Research on Assembly Language Course Based on Applied Talents Training

Teaching of 8088/86 Programming with 8086 Assembly Emulator

Study on the course-Assembly Language Programming

Analysis of the development of smart banking in the banking industry in recent years