Secure Personal Account Management Application in Assembly Language

CSE2006 Microprocessor and Interfacing
J-Component Report

Submitted by

Adi Thakkar - 19BCE0074

Dharmik Govani - 19BCE0091

Abhilash Yannam - 19BCE2180

in partial fulfilment for the award of the degree of

BACHELOR'S IN TECHNOLOGY

in

COMPUTER SCIENCE ENGINEERING

VIT®

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

FALL SEMESTER 2021-22
SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

Faculty: Dr Vikas Vijyavargiya

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have gotten this all along with the completion of our project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank Dr Vikas Vijayvargiya, for providing us with an opportunity to do the project and giving us all support and guidance, which made us complete the project on time and meet the set goals.

We owe our deep gratitude towards the honourable Chancellor, Dr G Vishwanathan and our institution VIT, VELLORE which took a keen interest in the project and guided us all along, till the completion of our project by providing all the necessary information for developing a good system.

Last but not least we place a deep sense of gratitude to our family and friends who have been constant sources of inspiration during the project.

TABLE OF CONTENTS

Abstract

In this project, we intend to develop a banking account system using EMU8086. The personal finances of an user can be handled using this project where a user can register their account, secure it with a pin and make transactions accordingly. This system will take care of all the transactions made by the user and alert the user if a transaction is possible or not. A Two - Level security authentication has been implemented that consists of a Pin and a Password. Moreover, the user can also recover their account using security questions. This software developed in Assembly Language helps users maintain their finances and is faster than many other existing applications.

Introduction

There exist many implementations of the bank account management system. But the issue is that these systems are difficult to implement and have a lot of communication overhead and a lot of server data is required which is costly to implement. Moreover, many times heavy, external softwares has to be downloaded which overloads the system. These are relatively new technologies compared to the assembly language programming and it will take time for software developers to put these technologies to good use. The backend/server side programming utilizes a single thread which requires heavy computing power, it lacks parallel processing.

We are proposing an implementation of the bank account management system in assembly programming of the 8086 microprocessor. It is simple, cost free to implement and easier to compile and test since it's all possible in one emulator. The assembly language can run many complex jobs in a simple manner, it is memory efficient, faster in speed and execution. It is hardware oriented and requires fewer instructions to get results. It is also easily portable to other assembly language emulators.

Literature Survey

The usage of macros in assembly language programming is discussed in this work. Following a brief overview of the basics, two applications in which macros have shown to be extremely useful are discussed. The ease with which structured programming techniques can be incorporated is emphasised in particular [4].

This paper's strategy for teaching assembly language programming has been demonstrated to be effective. Seneca College students have successfully constructed microprocessor-based designs that necessitated the use of complicated software. The resulting code, written entirely in assembly language, is succinct, easy to read and debug, and most significantly, it employs structured programming techniques [5].

Methodology

The system begins with the cover of the system shown in the Output section and provides user with the following options:

1. Create Account
2. Print Account Details
3. Withdraw Money
4. Deposit Money
5. Reset Account
6. Modify Account Details

1. Create Account

The first feature that is implemented is Create Account where a user is prompted to enter his Name, Pin, Password. The user is then asked to select a security question which he would like to answer in order to recover his account in case of a Password/Pin loss.

2. Print Account Details

The second feature that is implemented is Printing Account Details. When the user selects this feature, the user will be asked to insert their Pin and Password to authenticate the user. If the entered pin or password is wrong, the user will be sent back to the Home page. On entering the correct credentials, all the current account details will be printed that include Name, Pin, Password and current Account Balance.

3. Withdraw Money

The next feature implemented is Withdraw Money. When the user selects this option, the user will be asked to insert their Pin and Password to authenticate the user. If the entered pin or password is wrong, the user will be sent back to the Home page. On entering the correct credentials, the user will be provided with a price list of the amount that the user wants to withdraw from their account. Current list has 4 options namely Rs 1000, Rs 2000, Rs 5000 and Rs 10000. The user can select any one option of the above and if the account has sufficient balance, the amount will be deducted successfully.

If there is not sufficient amount in the account, the user will be prompted with a message “You Are Withdrawing Too MUCH !”.

4. Deposit Money

The next feature implemented is Deposit Money. When the user selects this option, the user will be asked to insert their Pin and Password to authenticate the user.   If the entered pin or password is wrong, the user will be sent back to the Home page. On entering the correct credentials, the user will be provided with a price list of the amount that the user wants to deposit into their account. Current list has 4 options namely Rs 1000, Rs 2000, Rs 5000 and Rs 10000. Selection of any one option from the above price list will result in deposition of the amount.

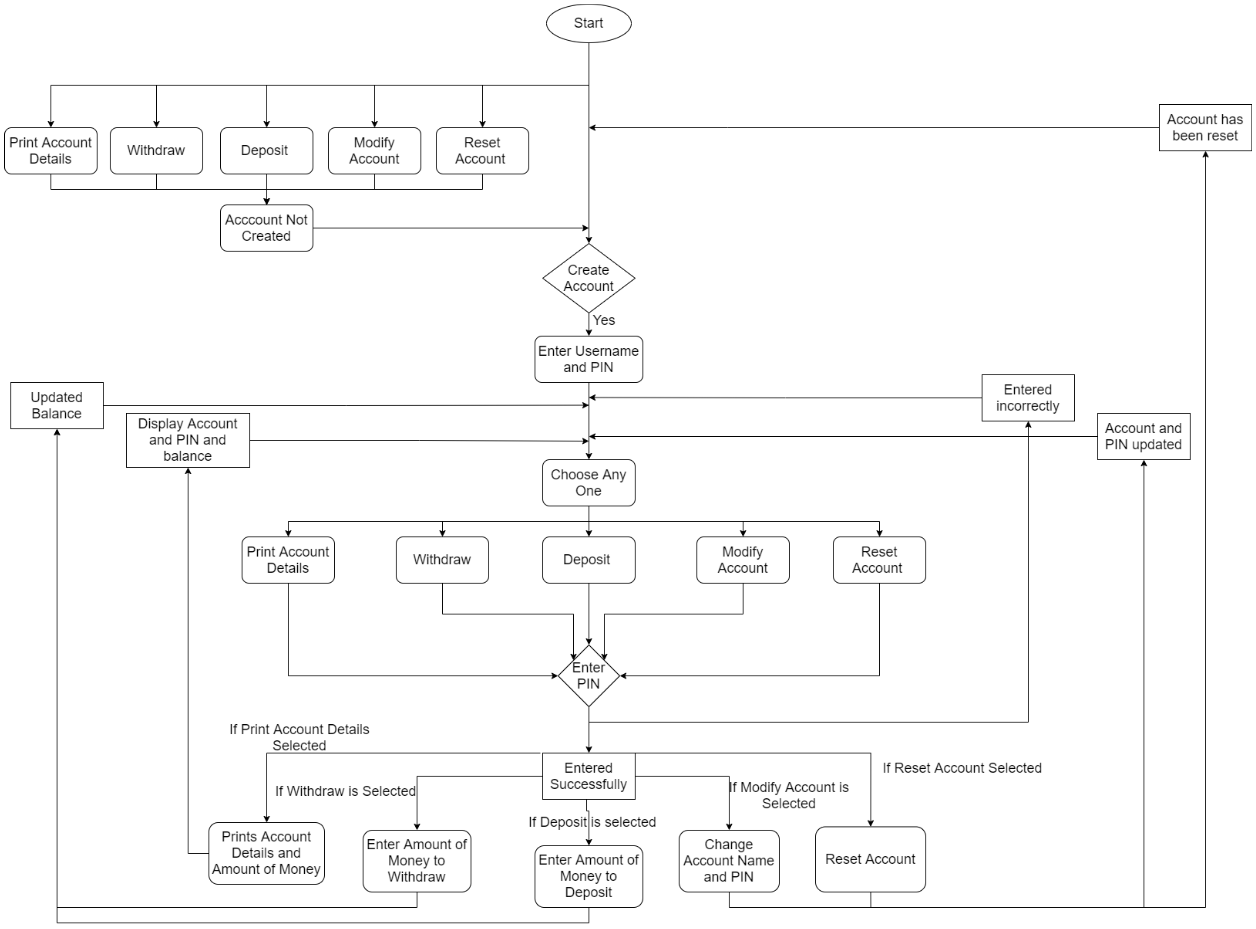
5. Reset Account

The next feature implemented is Deposit Money. When the user selects this option, the user will be asked to insert their Pin and Password to authenticate the user. If the entered pin or password is wrong, the user will be sent back to the Home page. On entering the correct credentials, the account managed by the software will be reset and a new account will have to be created to perform any such operations.

6. Modify Account Details

The second feature that is implemented is Printing Account Details. When the user selects this feature, the user will be asked to insert their Pin and Password to authenticate the user. If the entered pin or password is wrong, the user will be sent back to the Home page. On entering the correct credentials, the user will be asked to enter his new Name, Pin and Password. The user can enter the current details in any field, if he does not want to change that particular data. After this, all the details will be modified accordingly.

Flow Diagram



Proposed algorithms

The algorithms used to implement the system are the procedures and macros. The main procedures in our implementation are -

1) Asking PIN for every functionality-
    a) PIN count in CX and PIN in SI
    b) Checking the PIN count for ‘n’ amount of times.
    c) Comparing each keystroke with the pin number at index CX.
    d) Keep repeating until PIN is correct, if incorrect at any step, go to the main page.

2) Main process -

    At every step the user has to enter input to select whatever option they want to go forward with.

    a) JE used to verify which option was selected and jump to the respective procedure.
    b) JMP used to jump back to the main menu, if the user wishes to go.

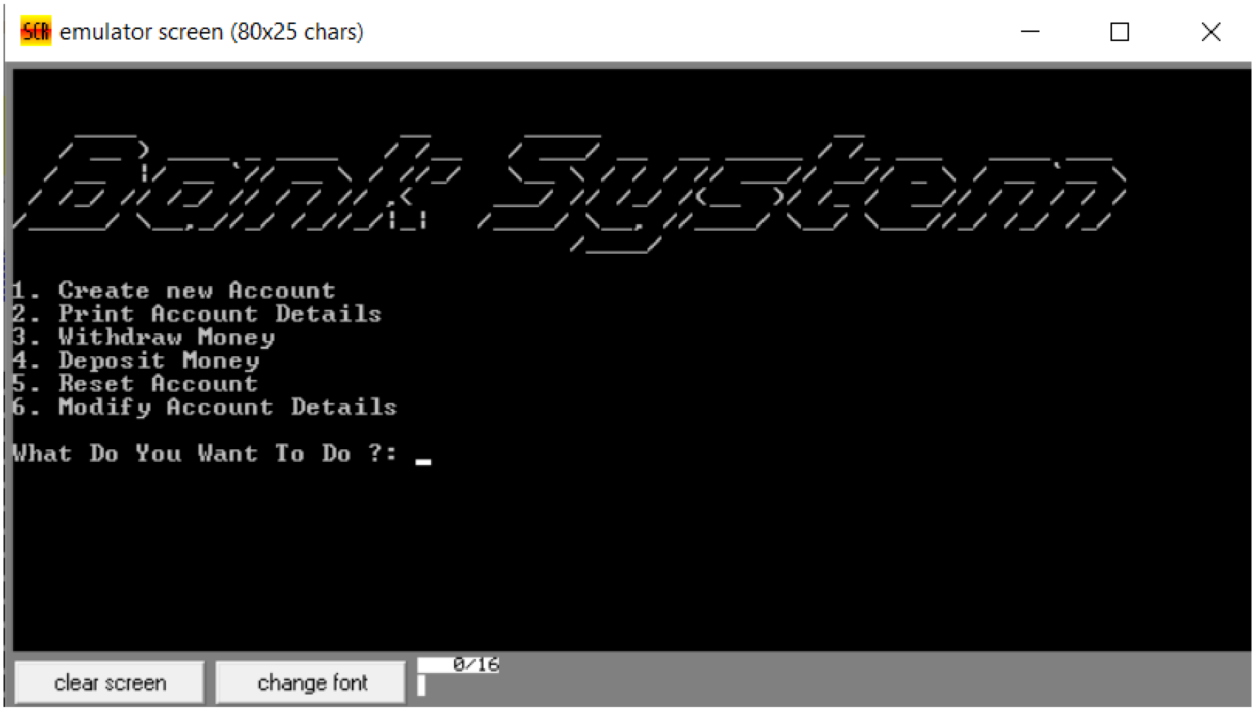
3) Check if account is created process -
    a) Compare PIN count to 0.
    b) If 0 accounts are not created and notify the user that the account has not been created.
    c) If not zero, do nothing.

There is also a set of macros defined in the implementation those are,

1) To get the user input for PIN every iteration.
2) To modify the PIN details when chosen.

Results and Discussion

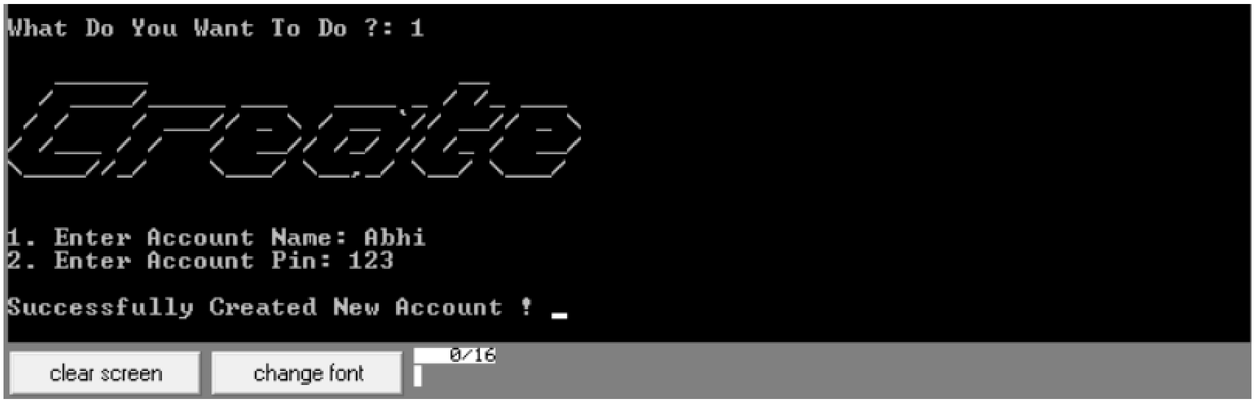
The main home screen of the bank account management system. The options of creating a new account, printing account details, withdrawing money, depositing money, resetting account, modifying account is given below.



```
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 ?: _
```

1. Create Account -

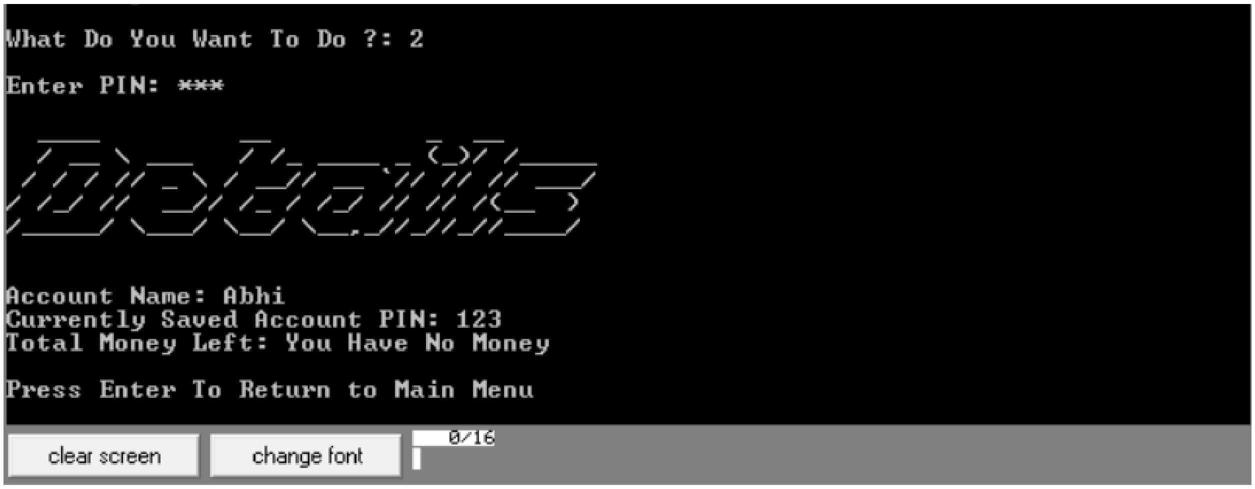
To create a new account we enter a name and PIN and then memory is created and these details are stored.



```
What Do You Want To Do ?: 1

1. Enter Account Name: Abhi
2. Enter Account Pin: 123

Successfully Created New Account ! _
```

2. Print account balance

Print the details of the bank account. Since I did not deposit any money the display shows “You Have No Money”.

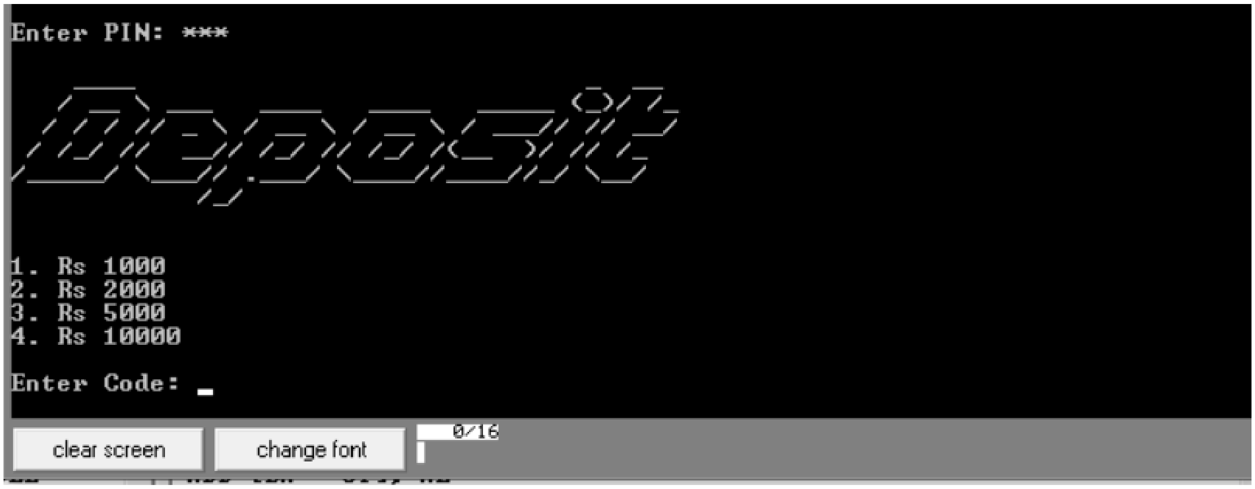


```
What Do You Want To Do ?: 2

Enter PIN: ***
```

```
Account Name: Abhi
Currently Saved Account PIN: 123
Total Money Left: You Have No Money

Press Enter To Return to Main Menu
```

3. Deposit Balance

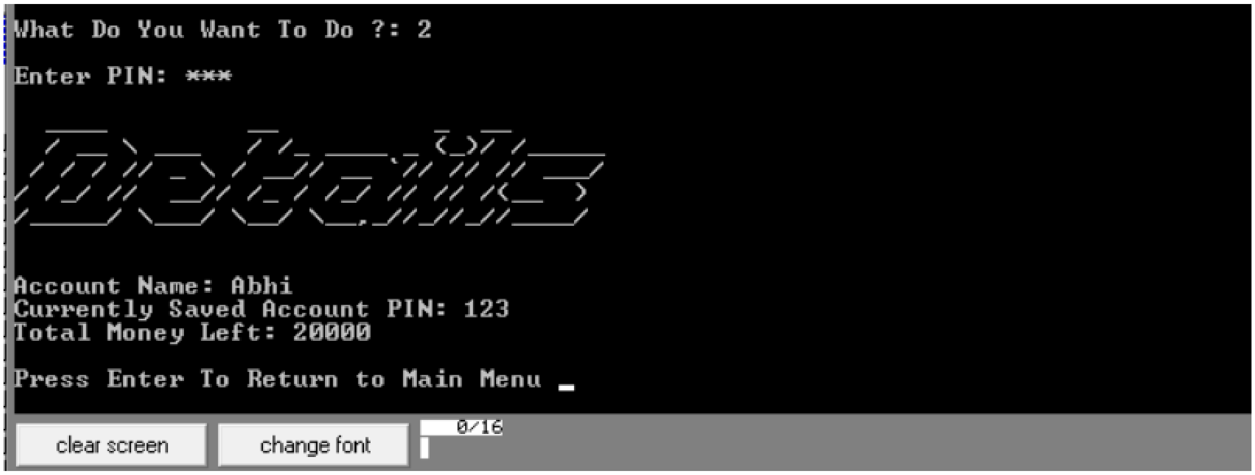
The options to deposit are 1000, 2000, 5000, 10000 and based on the option we choose that much amount will be deposited into the account.



```
Enter PIN: ***
```

```
1. Rs 1000
2. Rs 2000
3. Rs 5000
4. Rs 10000

Enter Code: _
```

After choosing option 4 two times the amount deposited is 20000 which is shown below

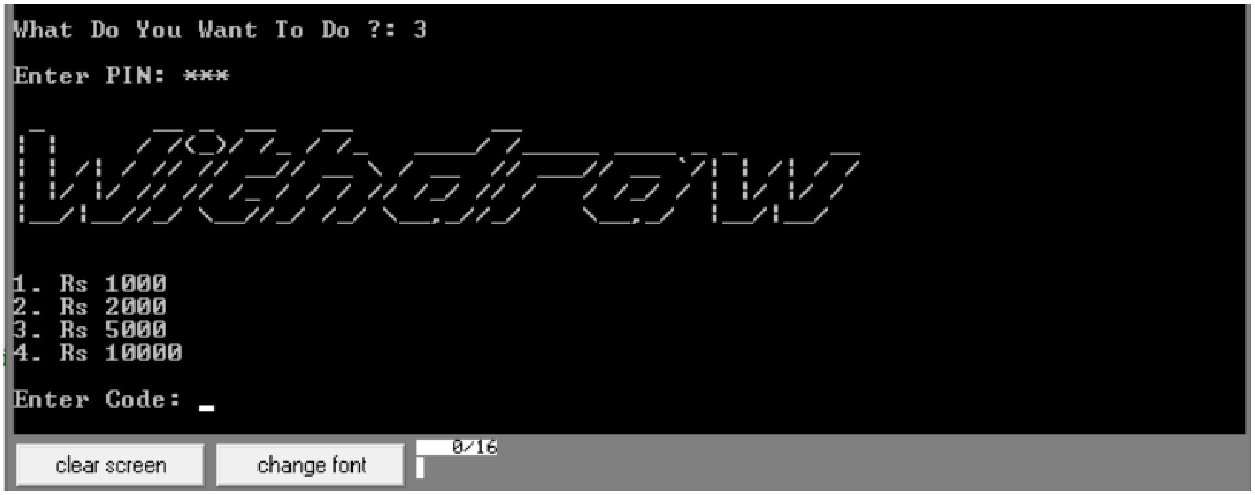


```
What Do You Want To Do ?: 2

Enter PIN: ***
```

```
Account Name: Abhi
Currently Saved Account PIN: 123
Total Money Left: 20000

Press Enter To Return to Main Menu _
```

4. Withdraw amount

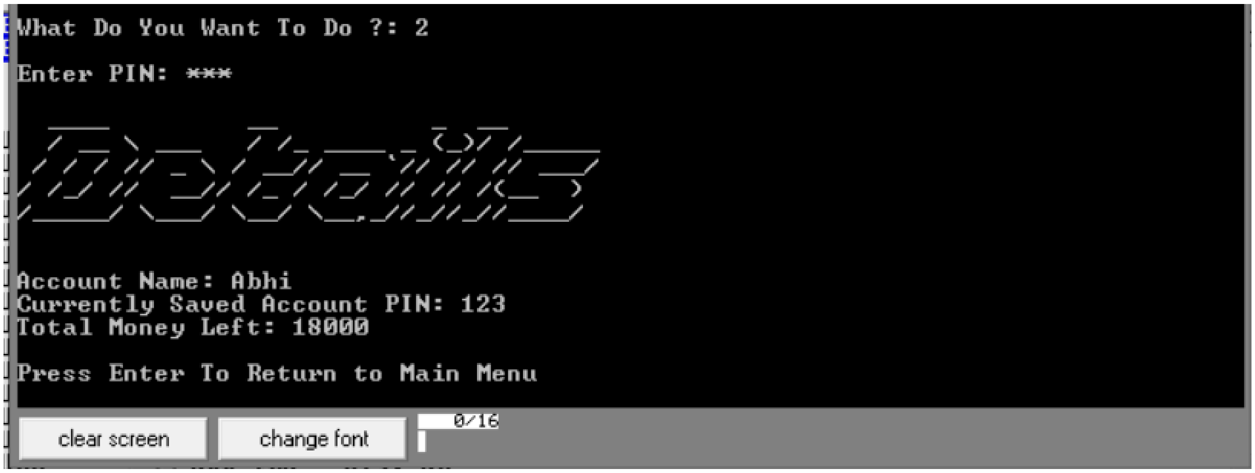
The withdrawal option can be chosen by pressing 3, we need to enter a PIN and the options of 1000, 2000, 5000, 10000 are shown below.



```
What Do You Want To Do ?: 3

Enter PIN: ***
```

```
1. Rs 1000
2. Rs 2000
3. Rs 5000
4. Rs 10000

Enter Code: _
```

Displaying the details and previously 20000 rs has been deposited now it has become 18000 rs.

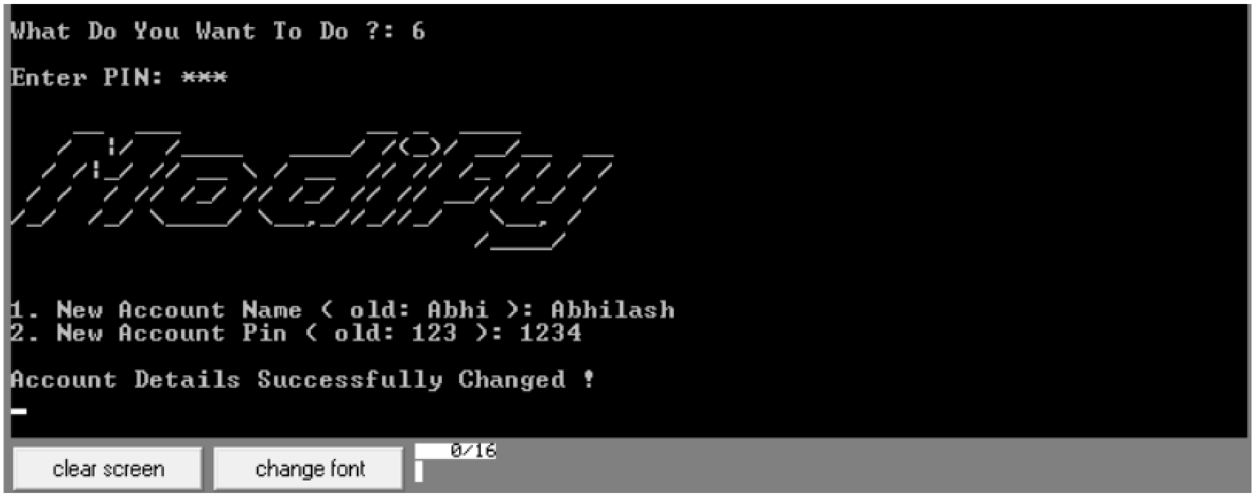


```
What Do You Want To Do ?: 2

Enter PIN: ***
```

```
Account Name: Abhi
Currently Saved Account PIN: 123
Total Money Left: 18000

Press Enter To Return to Main Menu
```

5. Modify Bank Account

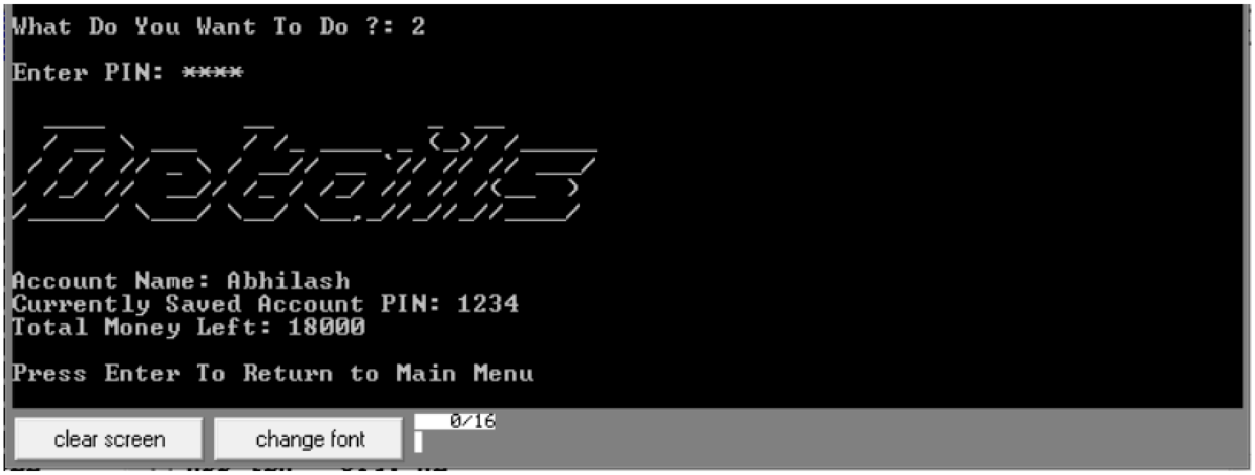
To modify the bank account we enter a new name and a new PIN number as given below and the details will be modified.



```
What Do You Want To Do ?: 6

Enter PIN: ***
```

```
1. New Account Name ( old: Abhi ): Abhilash
2. New Account Pin ( old: 123 ): 1234

Account Details Successfully Changed !
```

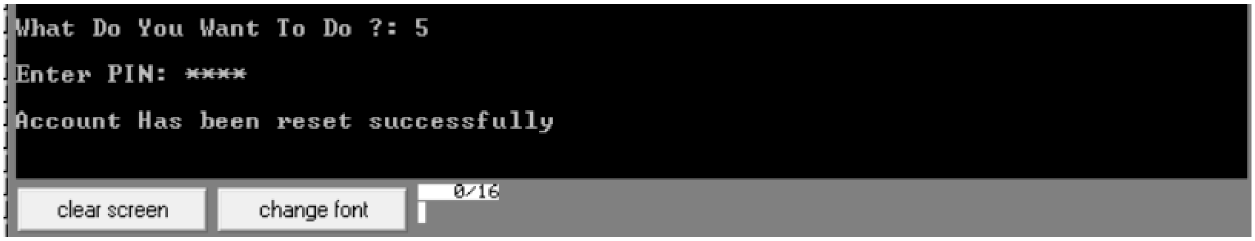
Printing account details and the changes have occurred according to what we expected.



```
What Do You Want To Do ?: 2

Enter PIN: ****

Account Name: Abhilash
Currently Saved Account PIN: 1234
Total Money Left: 18000

Press Enter To Return to Main Menu
```

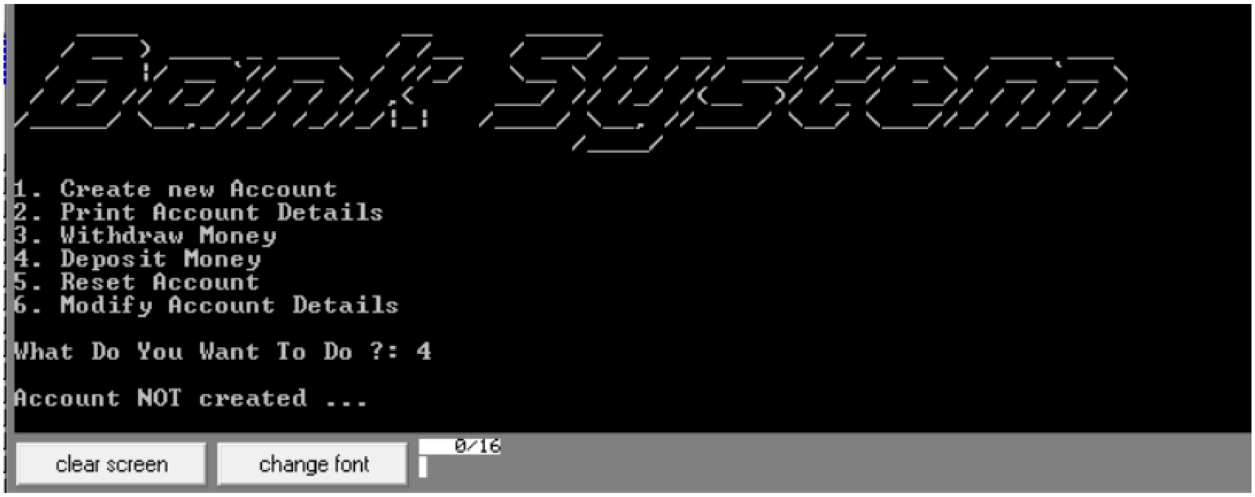
6. Reset Account

Clears all the stored bank account details from memory and requires PIN.



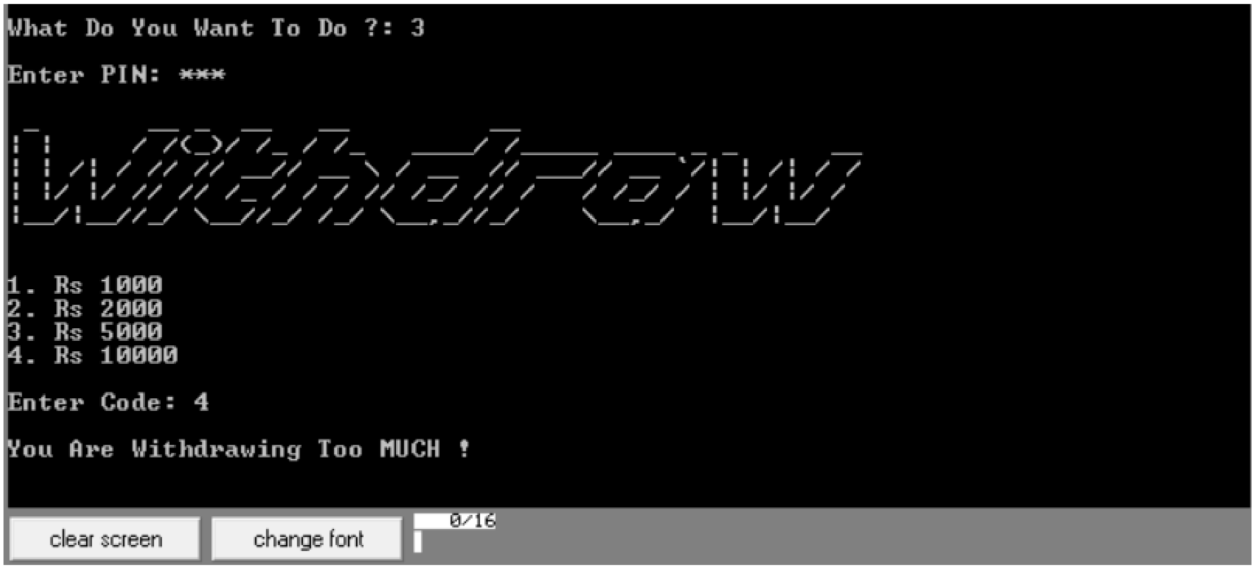
```
What Do You Want To Do ?: 5

Enter PIN: ****

Account Has been reset successfully
```

7. Extra features
    a. If a bank account is not created.



```
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 ?: 4

Account NOT created ...
```

    b. If money is being withdrawn is more than money deposited.



```
What Do You Want To Do ?: 3

Enter PIN: ***

1. Rs 1000
2. Rs 2000
3. Rs 5000
4. Rs 10000

Enter Code: 4

You Are Withdrawing Too MUCH !
```

Extra Features -
Dynamic PIN system, the length of the PIN can be of dynamic length.

Conclusion

Through the application we concluded that the 8086 microprocessor that the account management system was implemented with all the workflows and features.

1. **Increased security**: The user PIN is required before any transaction is made by the account or any edition is to be made to account details itself i.e change in name, PIN etc. , Hence provides additional security.

2. **Less chance of Human error**: This is connotation of the above feature, wherein inputting one's PIN before making any changes, makes sure that it was an intended step, and not a human/system error.

3. **Portability**- The code is written in ASM language, the lowest level of language, only next to binary language, Hence is easily portable from one device to another. Only requires an emulator to run.

4. **Optimised Code** - Due to the code being in ASM language, It is optimized in terms of the space required and the speed of execution. Also we have kept in mind the overheads in using various different methods in assembly language and have kept them to minimum.

Future Work

In the future we expect our program to be extended with these features -

1. **2 Factor Authentication** - The user has to enter a PIN and a PASSWORD, both of which if correct will give the access to the user. Thereby, increasing the security of the program.
2. **Forgot PIN feature** - In case the user forgets their PIN, We give a security question when they create an account, which when answered will allow them to set a new PIN .
3. **Savings plan-** Automated deduction system, which deducts a preset amount at regular intervals to pool in amount, the whole of which can be withdrawn at the end of the decided term with interest.

References

[1] Dandamudi, S. P. (2013). Introduction to assembly language programming: from 8086 to Pentium processors. Springer Science & Business Media.

[2] Giuma, T. A., & Randolph, T. (1995, March). Microprocessor tutorial using toolbook. In Proceedings of Southcon'95 (pp. 132-136). IEEE.

[3] Adams, T. L., & Zimmerman, R. E. (1989). An analysis of 8086 instruction set usage in MS DOS programs. ACM SIGARCH Computer Architecture News, 17(2), 152-160.

[4] Goodall, R. M. (1989). Macros: a neglected technique in assembly language programming. Microprocessors and Microsystems, 13(7), 437-444.

[5] MacKenzie, S. (1988). A structured approach to assembly language programming. IEEE Transactions on Education, 31(2), 123-128.