Campus	NPI
APU Foundation & Degree Programmes	

Please Tick (✓) for Re-do Assignment

Coursework Submission and Feedback Form

Name	RABIN ACHARYA			Name of Group Members (If Applicable)
Student No.	NPI000035	Intake:	NPIIF1909IT	
Module Code & Title	CT018-3-1 ITC			
Assignment Title	MALASYIAN URBAN BANK MANAGEMENT SYSTEM (MUBMS)			
Name of Lecturer	MR. SUSHIL ADHIKARI			
Date Due		Student E-Mail: Rabin7acharya@gmail.com		

I have read and understood the regulations on Plagiarism and Academic Dishonesty and declare that the work submitted does not breach those regulations. **Signed:** _____

You must hand in to the designated APU Administrator – ensure that you receive your receipt.

Received By		Signature		Date		Time	

Criteria	Weighting	Fail	Marginal Fail	Pass	Credit	Distinction
* Design Solution (20)	16	16				83 (A+)
* Coding (30)	25	25				
* Documantation (20)	16	12				79 A
* Demonstration (10)	9	9				
* question & Answer (20)	17	17				
	83/100	79				

Additional Comments: (These may be listed below or attached)

* good system design – flowcharts contain few or no mistakes at all.
* Coding is also good.
* C-pwg. concepts nicely explained.
* formating looks good. (Abstract & Acknowledgement should be in different page.)
* software demo was nice. answered well during demo of software
* Questions also

Provisional Assessment Result:

Grade	Date	Lecturer's Initial
A+	12/02/2021	

Taking account of above factors, the overall provisional assessment of your work is:

Distinction		Credit		Pass		
A+: 80-100%	A: 75-79%	B+: 70-74%	B: 65-69%	C+: 60-64%	C: 55-59%	C-: 50-54%
Marginal Fail		Fail				
D: 40-49%		F+: 30-39%		F: 20-29%		F-: 0-19%

The comments and assessment result are subject to both internal and external moderation at the appropriate Examination Board. Consequently, they may not reflect your final grade. You may not appeal against these result on ground of Academic Judgement.

A · P · U
ASIA PACIFIC UNIVERSITY
OF TECHNOLOGY & INNOVATION

ASSIGNMENT

INFOMAX COLLEGE OF IT AND MANAGEMENT

CT018-3-1

INTRODUCTION TO C PROGRAMMING

HAND OUT DATE: *11*/OCT/2019

HAND IN DATE: *23*/DEC/2019

WEIGHTAGE: *80%*

INSTRUCTIONS TO CANDIDATES:

1. Submit your assignment at the administrative counter.
2. Students are advised to underpin their answers with the use of references (cited using the Harvard Name System of Referencing)
3. Late submissions will be awarded zero(0) unless Extenuating Circumstances are upheld.
4. Cases of plagiarism will be penalized.
5. The assignment should be bound in appropriate style (comb bound or stapled)
6. Where the assignment should be submitted in hardcopy and softcopy, the softcopy of the written assignment and source code (where appropriate) should be on a CD in an envelope / CD cover and attached to the hard copy.
7. You must obtain 50% overall to pass this module.

C

ORIGINALITY REPORT

5%	1%	0%	5%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	1%
2	Submitted to Aspen University Student Paper	1%
3	Submitted to Punjab Technical University Student Paper	1%
4	Submitted to RDI Distance Learning Student Paper	1%
5	home.hit.no Internet Source	<1%
6	Submitted to University of Northumbria at Newcastle Student Paper	<1%
7	Submitted to Southern New Hampshire University - Continuing Education Student Paper	<1%
8	GANG QU, NAOYUKI KAWABE, KIMIYOSHI USAMI, MIODRAG POTKONJAK. "CODE	<1%

Abstraction

The study is based on the MUBMS (Malaysian Urban Bank Management System) which is a console framework based on C-Programming. The main purpose of this bank management system is to simulate users with simple and fast features to provide a realistic banking experience. This report is conceived with the flowchart and pseudocode. It was a great experience designing the application with the use of different C-Programming concepts. Using the function has helped to break the tasks into different functions while coding. Similarly, structure, array, pointer, variables and iterative constructs are the main c programming used in the project.

Acknowledgement

This project/report was generated with a lot of help and support. First of all, I would like to thank god for gracing my life with blessing and opportunities.

I would like to thank our module teacher Sushil Adhikari sir for providing continuous guidance and help. I would like to offer my special thanks to our college Infomax College of IT and Management and Asia Pacific University. I was helped by my colleges and other close friends. So, thank you all for your direct and indirect helping hands.

Contents

1. Introduction and Assumptions 3
2. Project Design 4
   2.1. Main Menu 4
   2.2. Login 6
   2.3. Admin Menu 8
   2.4. Customer Menu 10
   2.5. ID Auto generation 12
   2.6. Deposit 14
   2.7. Withdraw 16
3. Functions Used 18
   3.1. Library functions 18
   3.2. User defined functions 18
4. C programming concepts 21
   4.1. Functions 21
   4.2. Structure 22
   4.3. Array 22
   4.4. Loops 23
      4.4.1. For loop 23
      4.4.2. While Loop 24
   4.5. Decision-making 25
   4.6. File handling 26
   4.7. Pointer 26
5. Additional features 27
6. System Limitations 28
7. Sample Output 29
8. Conclusion 35
9. References 35

1. Introduction and Assumptions

This report is based on the Malaysian Urban Bank Management System which is a C Programming based console application. The main aim of this bank management system is to simulate users to provide a realistic functionality in MUBMS and use the application like real banking. This console application has functionality to keep user records, perform transactions, add/remove users and so on. Below are the types of users who can use the MUBMS.

a. Admin

Admin is the main user of the bank application having authority to see user records, manage them according to requirement.

b. Customer

The main purpose of creating this console application is to serve customers. So, customers are the also an integral part of this application. They are allowed to create an MUBMS account, perform transactional activities, see their information, update the saved information.

The app was designed for efficient and reliable banking experience to users. When a user signup into the system, they are provided with a unique ID number and account number. The system distinguishes each user according to their unique ID number. An admin can view all the user's confidential data including username and password. The system is assumed and designed in such way that a customer cannot withdraw amount more than available balance in their account. Similarly, the customer cannot deposit more than $1000000.00 per one transaction.

The application is lite and easy to use which helps customers for saving time, reducing travel cost. The system is designed for users to perform mobile banking experience and offer service at anytime, anywhere.

2. Project Design

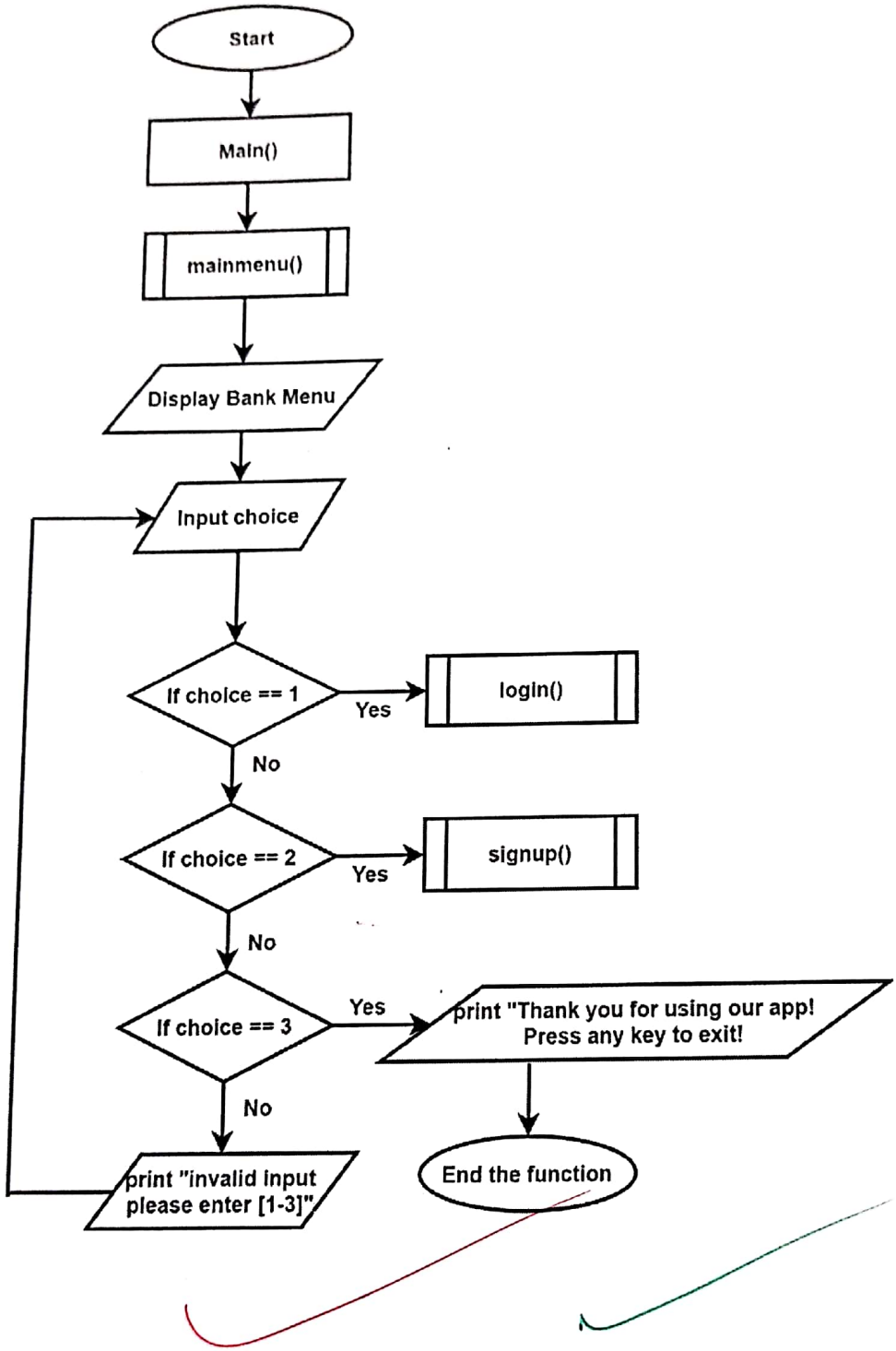
The mini project is often designed using pseudocode and flowchart for coding. The pseudocode is the line by line description of a program whereas, flowchart is the pictorial representation of pseudocode and algorithm. The pseudocode and flowchart below show the in-depth design of the entire program.

2.1. Main Menu

Pseudocode:

1. Start the program.
2. Display Main Menu of the Bank application.
3. Input choice.
4. Switch the choice entered.
   If choice == 1, then go to Login menu,
   If choice == 2, then go to Signup menu,
   If choice == 3, then exit the program,
   Else print "Invalid Input, please enter [1-3]",
   Return to step 3,
5. End the If statement,
6. End the function.

Flowchart:

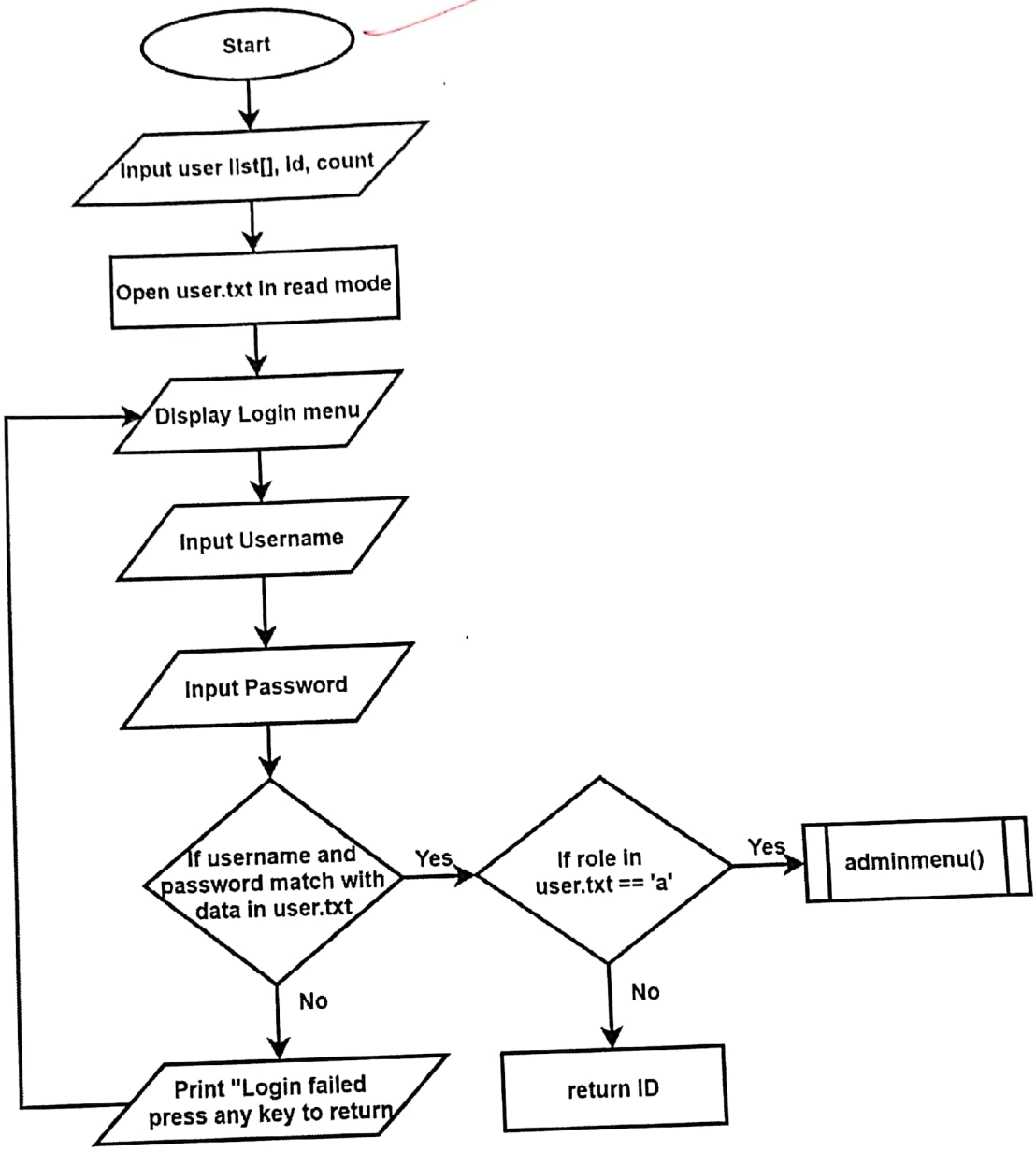


2.2. Login

Pseudocode:

1. Start the program,
2. Input user list[], id, count,
3. Open user.txt in read mode,
4. Display Login Menu,
5. Input user name and password,
6. If username and password match with data in user.txt,
   If role in user.txt == 'a',
   Go to adminmenu() function,
   Else,
   Return id,
   Close user.txt,
   Else,
   Print "Login failed, press any key to return",
   Go to step 4,
7. End the if statement,
8. End the function.

Flowchart:

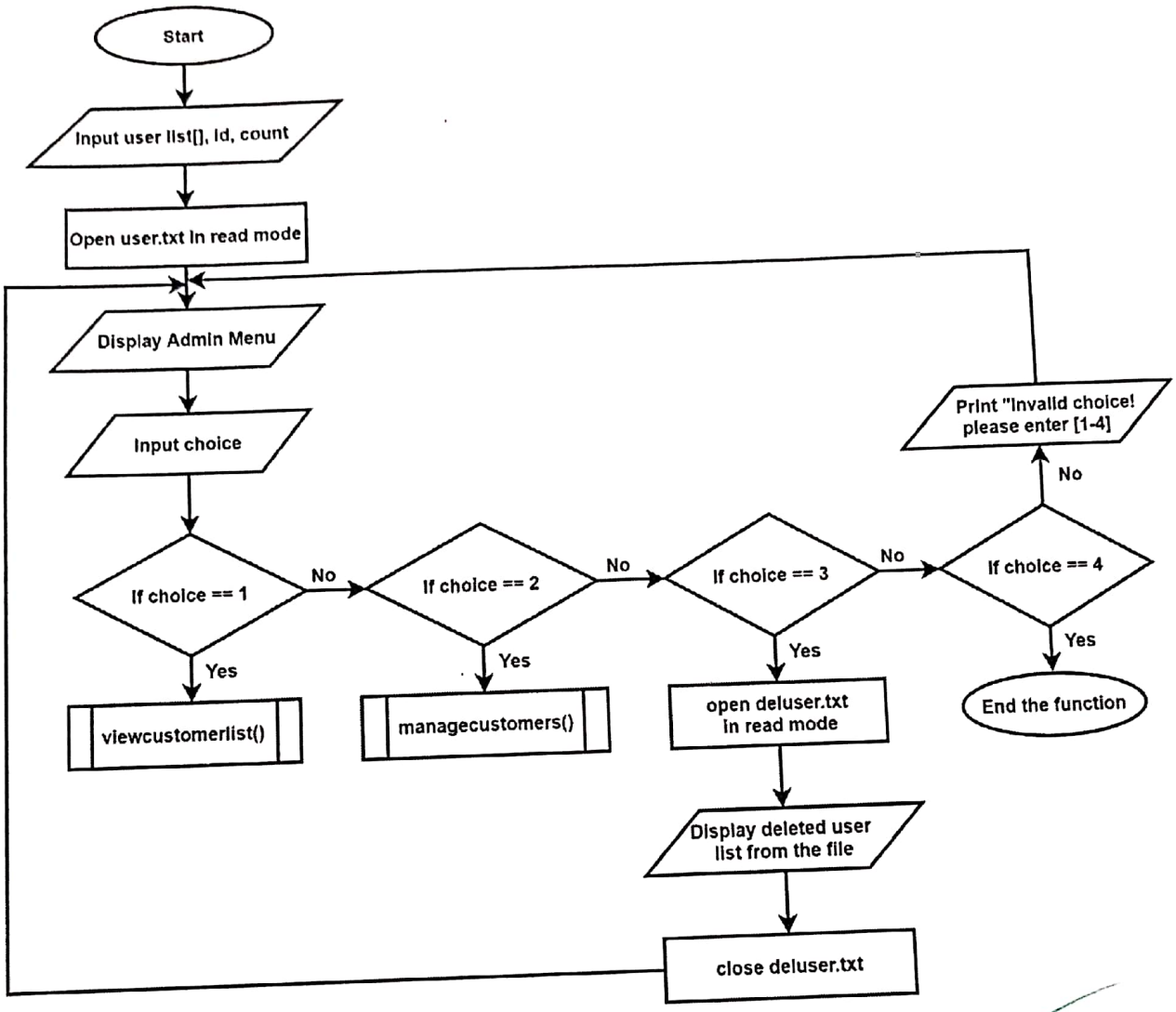


2.3. Admin Menu

Pseudocode:

1. Start the program,
2. Input user list[], count, id,
3. Display Admin Menu,
4. Input choice,
5. If choice == 1,
   Go to viewcustomerlist() function,
6. If choice == 2,
   Go to managecustomers() function,
7. If choice == 3,
   Open deluser.txt in read mode,
   Display removed user list,
   Close deluser.txt,
8. If choice == 4,
   End the function,
9. Else,
   Print "Invalid choice, please enter [1-4],
   Go to step 1,
10. End the if statement,
11. End the program.

Flowchart:

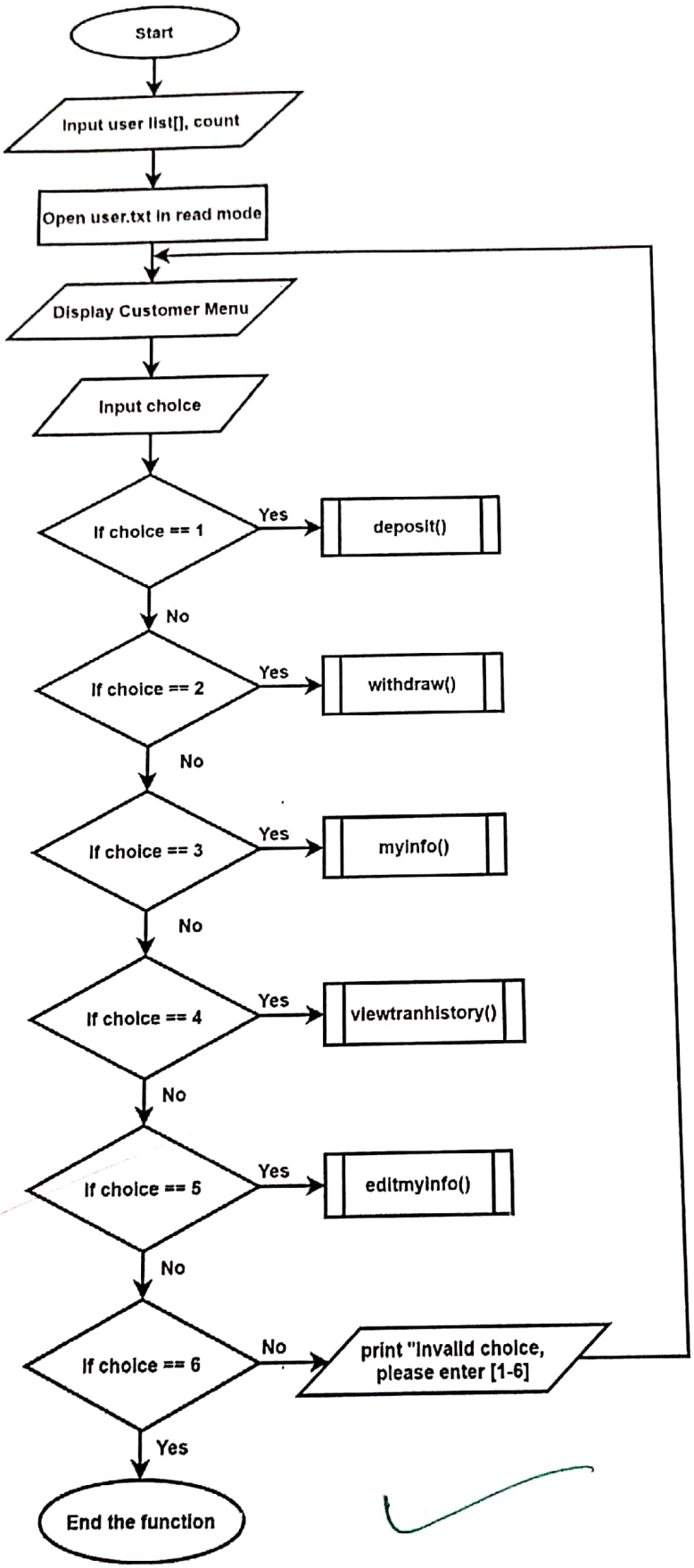


2.4. Customer Menu

Pseudocode:

1. Start the program,
2. Input user list[], count,
3. Display Customer Menu,
4. Input choice,
5. If choice == 1,
   Go to deposit() function,
6. If choice == 2,
   Go to withdraw() function,
7. If choice == 3,
   Go to myinfo() function,
8. If choice == 4,
   Go to viewtranhistory() function,
9. If choice == 5,
   Go to editmyinfo() function,
10. If choice == 6,
    Exit the function,
11. Else,
    Print "Invalid choice, please enter [1-6],
    Go to step 1,
12. End the if statement,
13. End the program.

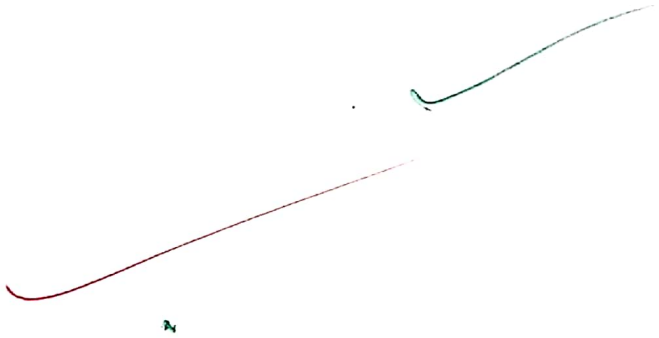
Flowchart:



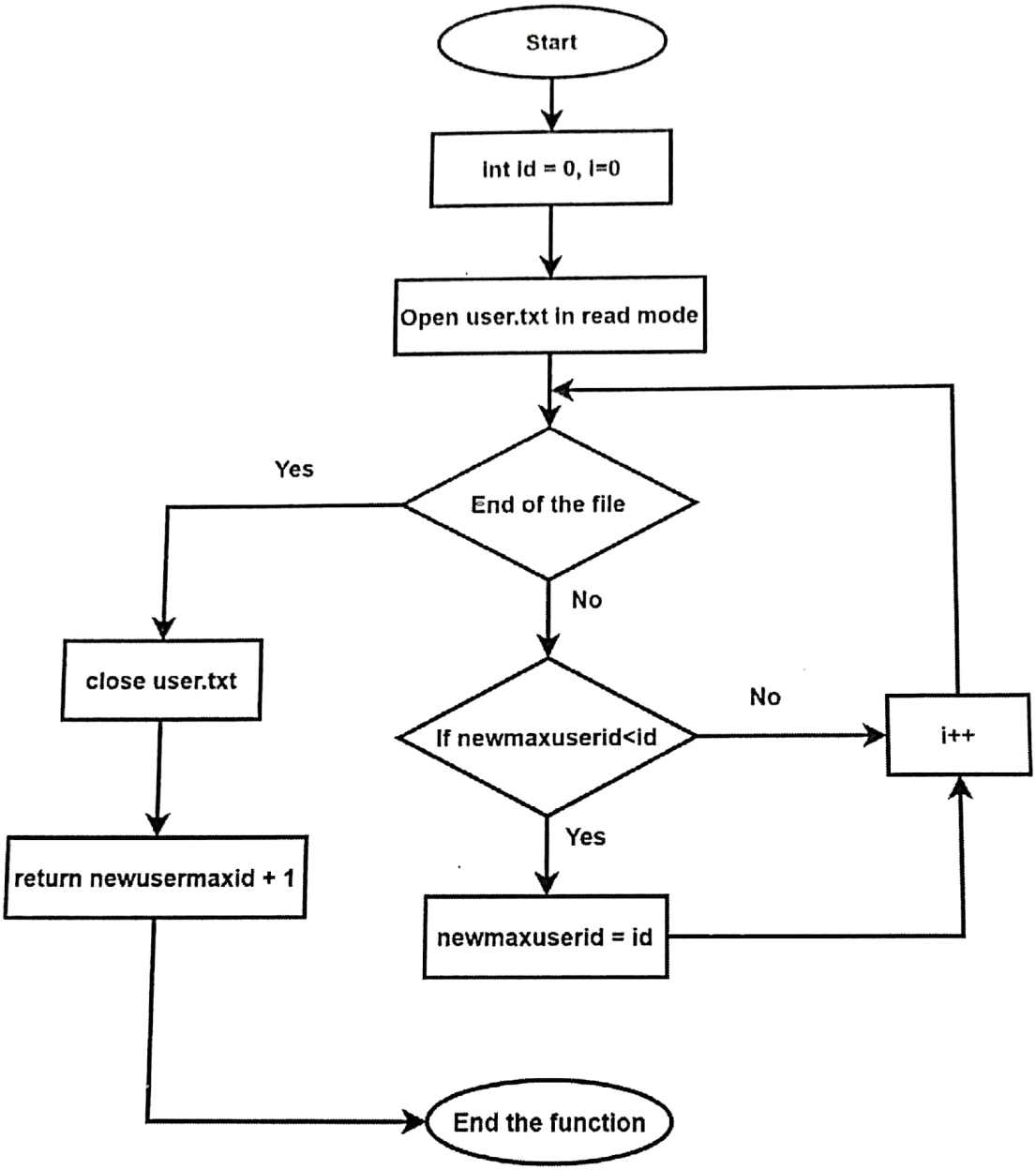
2.5. ID Auto generation

Pseudocode:

1. Start the function,
2. Initialize id = 0,
3. Open user.txt in read mode,
4. If newmaxuserid< id,
   Newmaxuserid = id
5. Close transaction.txt
6. Return newmaxuserid+1
7. End the function



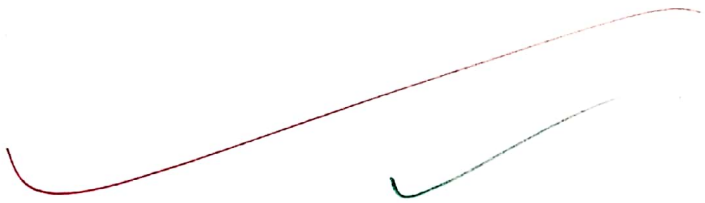
Flowchart:



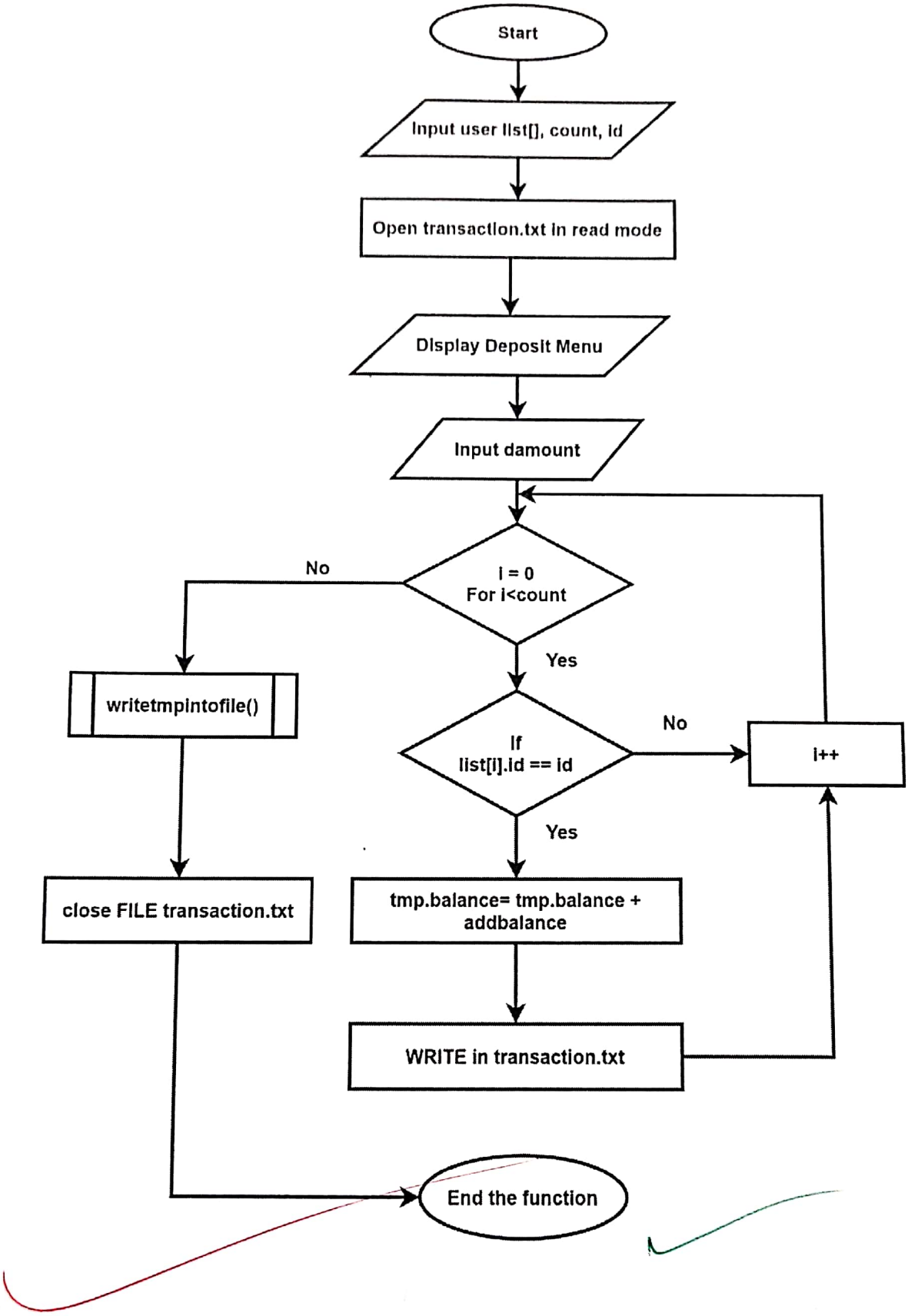
2.6. Deposit

Pseudocode:

1. Start,
2. Input user list[], count, id,
3. Open transaction.txt in append mode,
4. Display Deposit Menu,
5. Input addamount to deposit,
6. If tmp.id == id,
   Tmp.balance = tmp.balance + addbalance
   Write transaction into file,
7. Call writetmpintofile() function,
8. Print "addamount has been deposited"
9. End the function.



Flowchart:



2.7. Withdraw

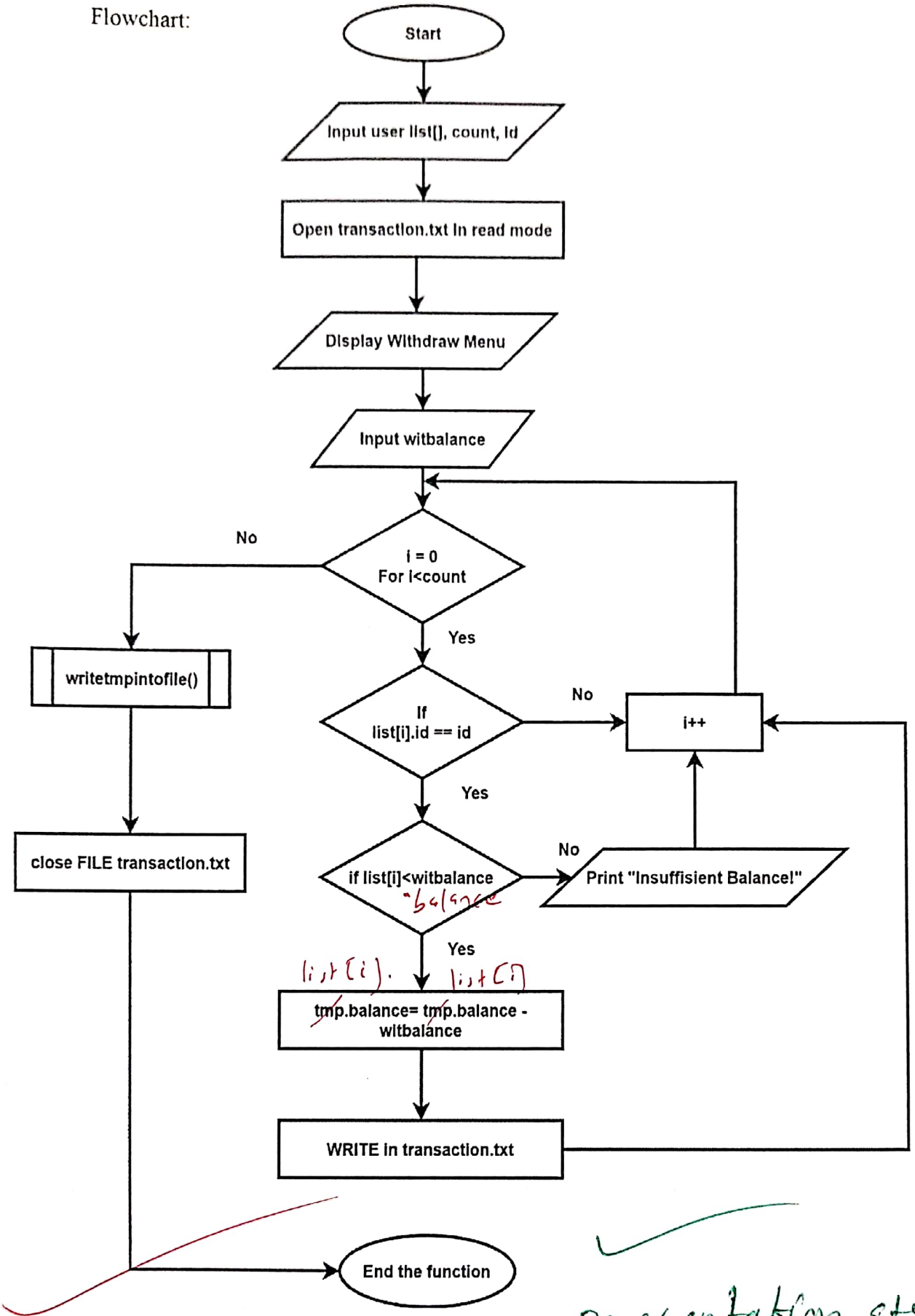
Pseudocode:

1. Start,
2. Input user list[], count, id,
3. Open transaction.txt in append mode,
4. Display Withdraw Menu,
5. Input witamount to withdraw,
6. Start for loop,

```
Tmp = list[i],
If tmp.id == id,
        If tmp.balance<witbalance,
                Print "Insuffient balance!"
        Else,
                Tmp.balance = tmp.balance - witbalance
                Write transaction into file,
                Print "addamount has been deposited"


End if

List[i]=tmp

End for
```

7. Call writetmpintofile() function,
8. End the function.

Flowchart:



presentation style and design looks good.

3. Functions Used

3.1. Library functions

The library function also called header files are used in this console application. The library functions along with their purpose are illustrated in the table below.

S.N.	Function Name	Purpose
1	#include	Used for shaping preprocessor to read in a text file.
2	<conio.h>	Used for creating a console platform screen
3	<stdio.h>	Used for input and output operations (printf and scanf).
4	<string.h>	Used for character array manipulation.
5	<windows.h>	Used for accessing system date in the program.
6	#define ENTER 13	Used for password hide. (ch==\0)

3.2. User defined functions

i. mainmenu()

It is the outer menu which consists functions like login, signup and exit.

ii. login()

This function allows users to log in with their username and password which they get form signing up in the application. After entering username and password they redirected to either admin menu or customer menu according to their user account type.

iii. signup()

This function enables users to create an MUMBS account as a customer. After completing signup process, the users are allowed to login with their newly created account.

iv. adminmenu()

The admin menu function is comprised with functions like viewuserlist(), manageusers().

v. viewcustomerlist()

This function allows admin to see all the user account created in the MUMBS including admins and customers.

vi. managecustomers()

This function consists of two major functions, i.e. removeuser() and makeadmin().

vii. removeuser()

Admin can remove the existing MUMBS accounts through this function.

viii. makeadmin()

Admin can assign admin role to other user accounts.

ix. customermenu()

In this function the customers can perform transactional activities like deposit withdraw from their respective account number. Moreover, they can edit their personal data, see their transaction history.

x. deposit()

Customer can deposit certain amount of money according to their desire and need.

xi. withdraw()

This function allows customer to withdraw amount which they deposited earlier in the MUMBS application.

xii. myinfo()

Through this function customer can see their personal details like name, address, account number, user ID, date of birth, phone number, existing balance which they provided earlier during signup.

xiii. viewtranhistory()

The customers can see all the transaction log associated with their account number since they created the bank account.

xiv. editmyinfo()

Customers can edit and save their previously stored information like name, address, phone number, username and password through this function.

4. C programming concepts

C programming language is one of the powerful and general-purpose programming languages. It is simple, flexible, structured programming language due to which it is popular and widely used. A program is divided into various modules which is suitable for testing, maintaining and debugging. There are lots of c programming concepts available for compiling a source code. Iterative functions, multiple conditional statements, data structures are the most common concepts in c programming language. Below are the concepts used while writing the code for MUBMS.

4.1. Functions

In c programming language, functions are of two types i.e. library functions and user defined functions. Here in this report, both library and user defined functions are used. The use of these function has served this project in many ways. Using function, it was easy to assign particular task to a particular function. It made debugging easier. Similarly, problem plotting was effective and efficient. Below are the functions used in the source code of MUBMS application.

Function sample:

```
void mainmenu()
{
  int choice, count;
  struct user list[50];
  count = loadaccountlistfromfile(list);
  system("cls");
  printf("\n\t\t**************************\n\t\t**** WELCOME TO MUBMS ****\n\t\t**************************");
  printf("\n\n\t1. Login");
  printf("\n\t2. Signup (for Customers)");
  printf("\n\t3. Exit");
  printf("\n\n\tWhich operation to perform?\n\t");
  scanf("%d", &choice); fflush(stdin);
  switch(choice)
  {
    case 1: customermenu(list, count); getch(); break;
    case 2: signup(); break;
    case 3: system("cls");
            printf("\n\n\tThank You for using our app!\n\tPress any key to Exit!\n"); return; break;
    default: system("cls");
             printf("\n\n\tInvalid Choice, Please Enter [0-3]\n\n\t"); getch();
  }
  main();
}
```

4.2. Structure

A structure is a user defined data type which may consist a group of different data types into a single structure. To create a structure a keyword 'struct' is used before the structure name. Below is the example of structure named user consisting different variable names different data types.

Sample structure:

```
struct user
{
    int id;
    long int acc_num, phone;
    float balance;
    char  acc_type, name[50], address[50], gender[6], username[20], password[20], role;
    struct date dob;
};
```

4.3. Array

An array is a data structure that consists a group of elements. It is a collection of one or more value of the same type. In this project, array is used for mainly two purpose i.e. storing user list, array of character.

Sample array:

Struct user list[];

char name[100], username[20], password[20], repassword[20], address[50], gender[6];

4.4. Loops

A loop is sequence of instruction that repeats until a specified condition is reached. The same question is asked again and again until no further action is required. In the bank management system application different loops are used like for loop, while loop.

4.4.1. For loop

Sample For Loop:

```
void removeuser(struct user list[], int count)
{
    int id, i;
    printf("%d", count);
    printf("Enter an ID to remove: ");
    scanf("%d", &id);
    for (i=0; i<count; i++)
    {
        if (id == list[i].id)
        {
            for (i=i+1; i<count; i++)
            {
                list[i-1] = list[i];
            }
        }
    }
    count--;
    writetmpintofile(list, count);
}
```

In the above function removeuser(); for loop is used for multiple time iteration. The loop iterates until the count I becomes equal to count.

4.4.2. While Loop

Sample While loop:

```
while(1)
{
    hide=getch();
    if(hide==ENTER)
        {
            p[i]='\0';
            break;

        }
    else
    {
    p[i]=hide;
    printf("*");
    }
        i++;
}
```

In the above program the iterates infinite times unless the user presses the enter button. The aim of this program is to hide the input characters with '*' until 'enter' button is pressed.

4.5. Decision-making

The decision-making statements in programming language decides the direction of flow of program execution. If-else statement, switch statements are used in this project to make some decisions according to the reuirements.

Sample- If-else statement

```
if(strcmp(un, usern) == 0 && strcmp(p, pw) == 0 )
{

    //user type verification
    if(role=='a')
    {
        adminmenu(list, count, id);
        return -1;
    }
    else
    {
        return id;
    }
    fclose(fp);
    return 0;
}
```

In the above if statement, a character type role variable is compared with character 'a'. If the condition satisfies, then adminmenu() function is called otherwise it returns id.

4.6. File handling

File handling is a process to create a data file, write data into the created file and read data from the respective data file. This project uses file handling as saving data into a txt file and the data is red, written and overwritten according to the requirement. There are three txt file in this project named 'user.txt' for storing the user information like name, address, account number, balance, date of birth and another txt file named 'transaction.txt' stores all the user performed transactional activities like deposit, withdraw with the date. Similarly, 'deluser.txt' stores the user data of removed users from the system by an admin.

Sample File Handling

```
void writetmpintofile(struct user list[], int count)
{
    int i;
    struct user tmp;
    FILE *fp=fopen("user.txt", "w");
    for (i=0; i<count; i++)
    {
        tmp = list[i];
        fprintf(fp,"%d %ld %s %s %s %s %d %d %d %s %ld %f %c\n", tmp.id, tmp.acc_num, tmp.name, tmp.username, tmp.password, tmp.gender,
        tmp.dob.day, tmp.dob.month, tmp.dob.year, tmp.address, tmp.phone, tmp.balance, tmp.role);
    }
    fclose(fp);
}
```

4.7. Pointer

A pointer is a variable whose value is the address of another variable. In this project, file pointer is used to handle the data structure form the txt file.

Sample pointer

```
FILE *fp;
fp = fopen("user.txt", "r");
```

5. Additional features

Though this console-based application is based on the minimum requirement of the assignment, some of the additional features have been used in the system to make it more realistic and efficient to use. Some of the additional features used in the MUBMS application are listed below:

i. Hidden Password
The user input password in the application is hidden using '*' symbol. This process is used in login and signup function.

ii. Auto-sync Removed User Information
After an admin removes a particular user from the bank system, the removed user's information is saved in a text file named 'deluser.txt' and the data can be viewed by the admin.

iii. Clean and User-Friendly UI
Though the application does not contain any graphics, the User Interface is clean, visually pleasing and friendly to all sorts of users.

6. System Limitations

The MUBMS is compiled using C-programming language and it is a console-based application, so the application lacks security and functionality. Some of the limitations of the MUBMS are listed below:

i. Poor Data Validation

Data validation is missing in the application and due to which the program can face many serious troubles. The file data can get corrupted if the invalid input is given from the user.

ii. Limited functionality

The application lacks many must have functions like balance/fund transfer facility, there is no interest available for deposited amount. Similarly, only few user information is collected form the user. Information like mail address, postal code is not included in the MUMBS application.

iii. Low Data Security

The system lacks database protection. All the user information including username and password are saved using file handling method. The file containing user information can be easily modified and deleted.

7. Sample Output

The figures below are the screenshots taken as sample output of the MUBMS application.

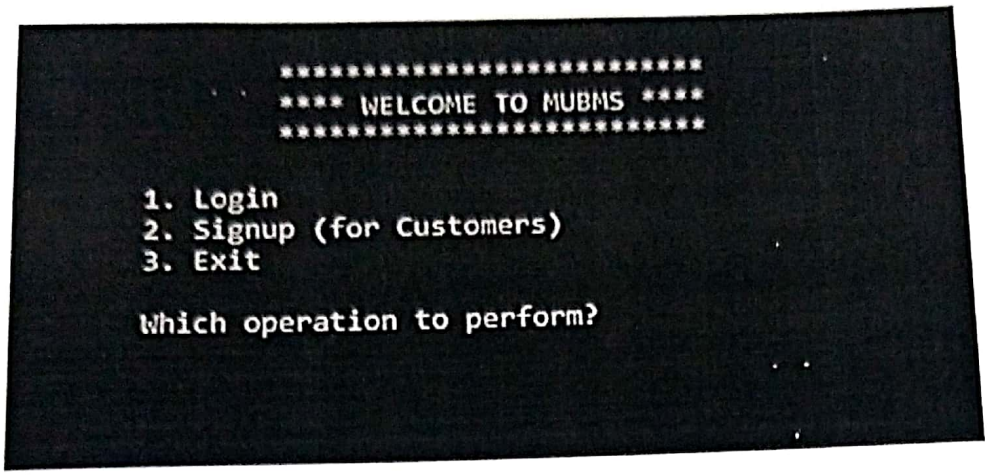


Figure 1: Main Menu Sample output

In the above figure, it is shown the main menu of MUBMS. There are 3 options available for user for Login, Signup and Exit the program.

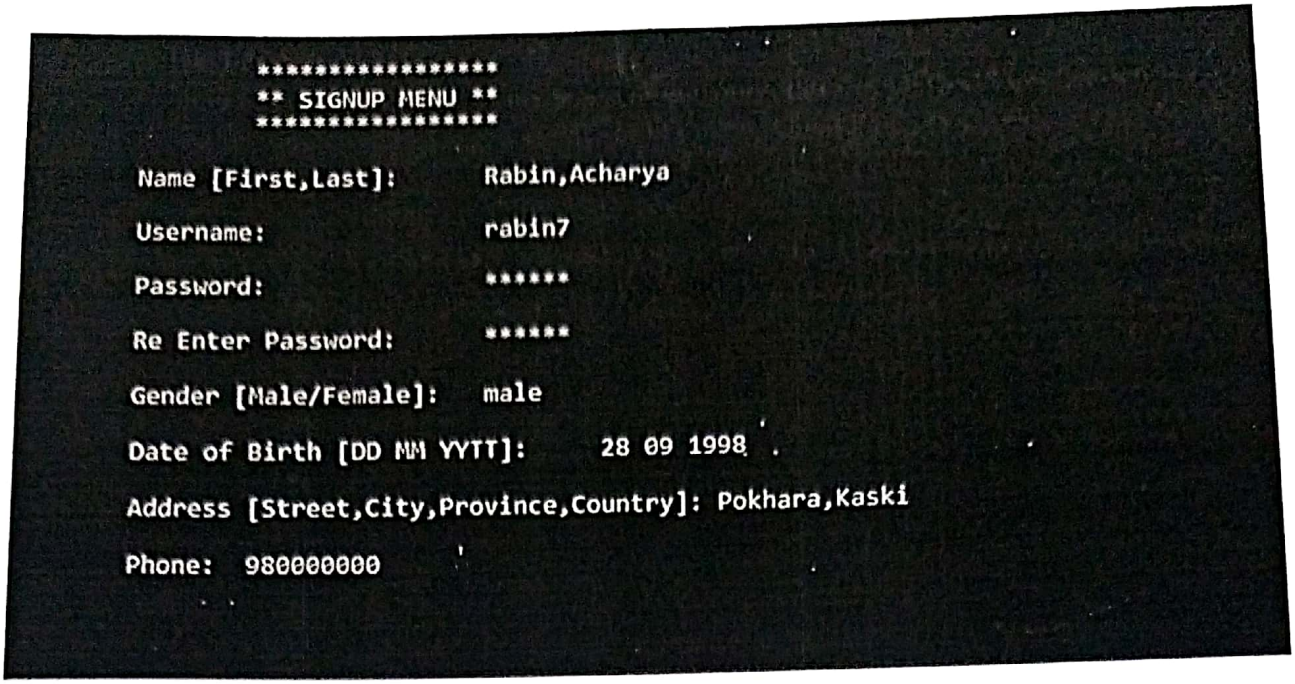


Figure 2: Signup Menu Sample output

In the Signup menu user has to enter the information as asked in the figure above. After completing this process, now the user can login with their newly created MUBMS account.

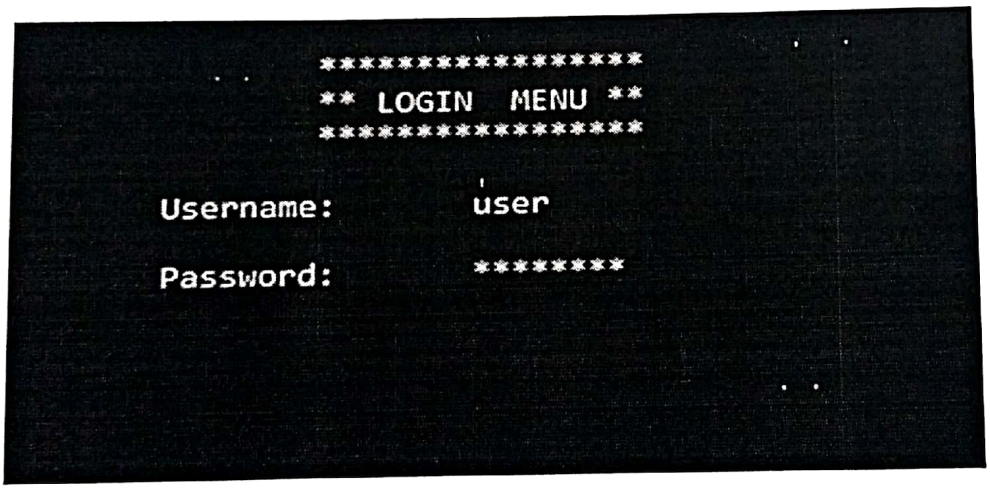


Figure 3: Login Menu Sample output

The above figure shows a user is asked to enter their login username and password which they get from signing up. After entering the correct username and password, the user can proceed to use the different functionality available ahead.

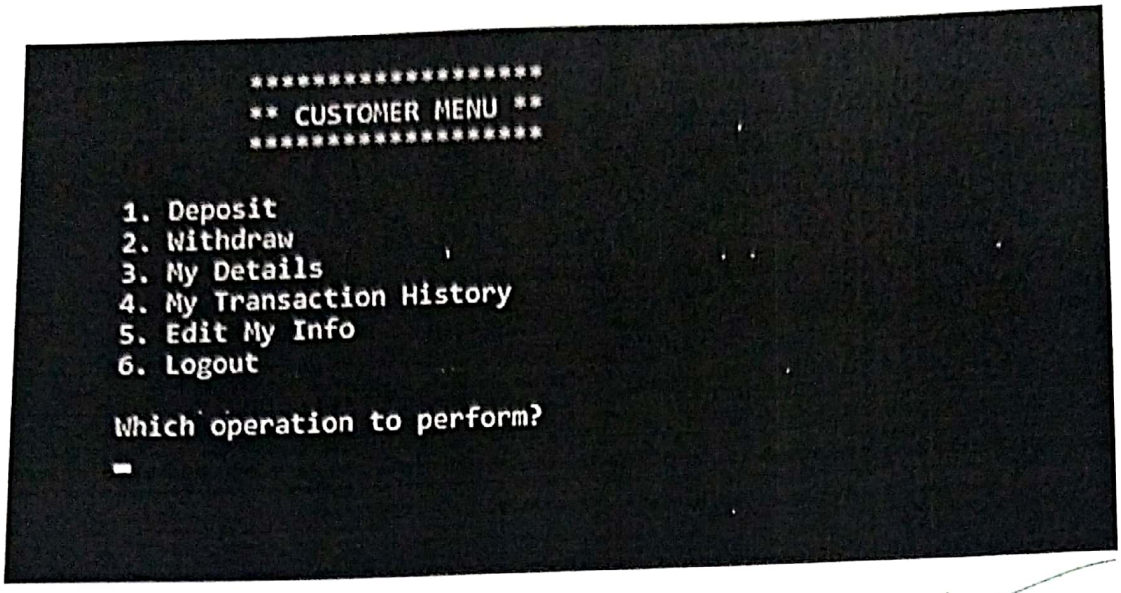


Figure 4: Customer Menu Sample output

In customer menu, a user is asked to choose a task among 6 different menus. Deposit, withdraw, see personal details, see transaction history, edit information and logout are the options available for user.

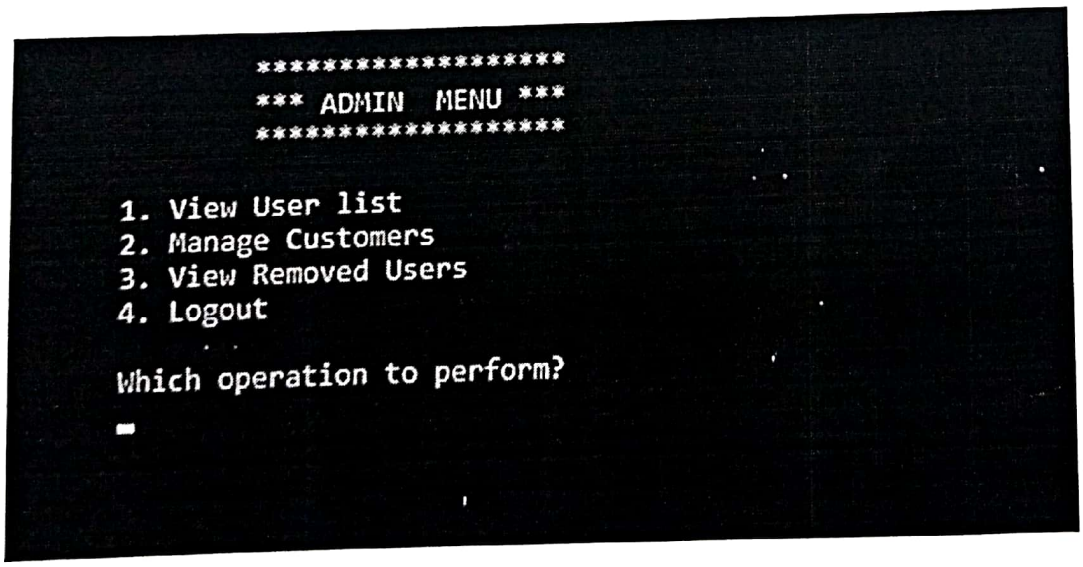


Figure 5: Admin Menu Sample output

A list of four function is displayed in Admin Menu. User can input the choice he/she likes to execute and proceed further for the displayed actions.

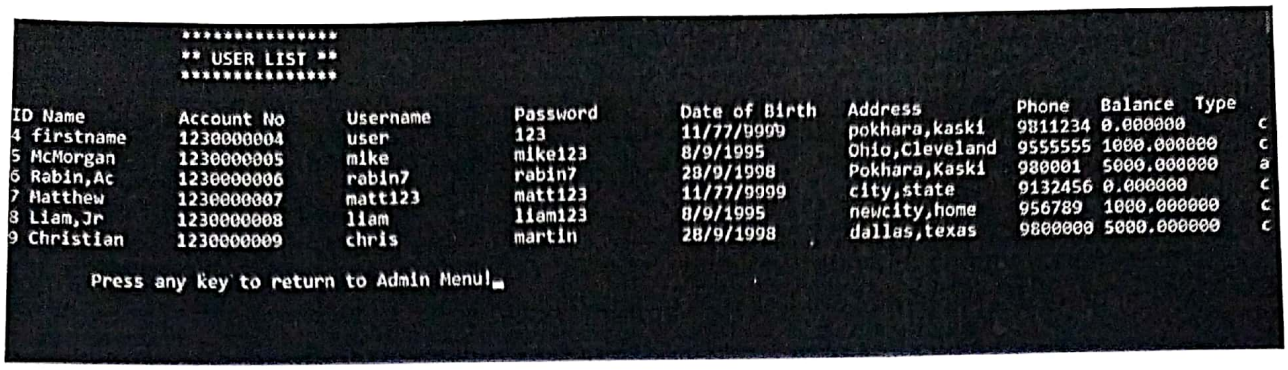


Figure 6: User List Sample output

When an admin chooses view user list function, the list is previewed as shown in the figure above. The users are sorted according to their ID number. The admin gets to see the details like name, address, username, password, date of birth, phone number, balance etc.

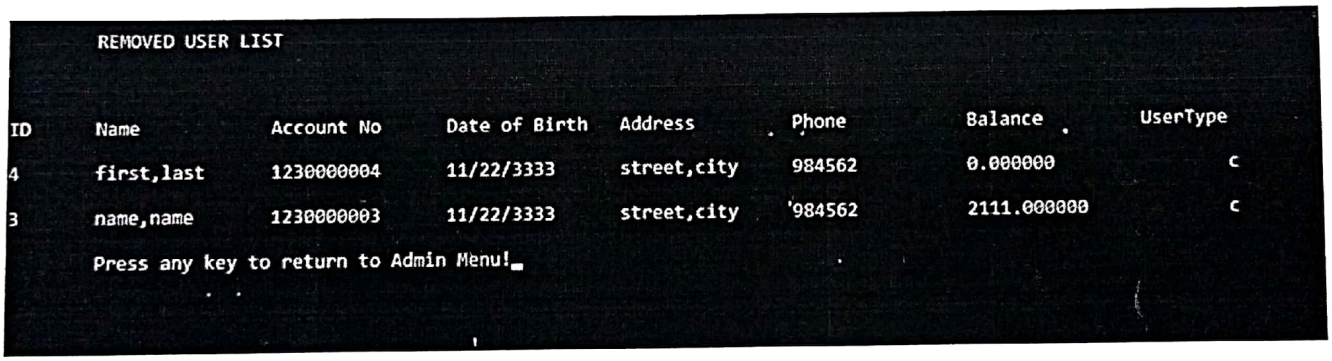


Figure 7: Removed User List Sample output

The removed user's list is displayed as in the above figure. They are sorted according to the time they were deleted. The newly deleted user is listed at the bottom of the list.

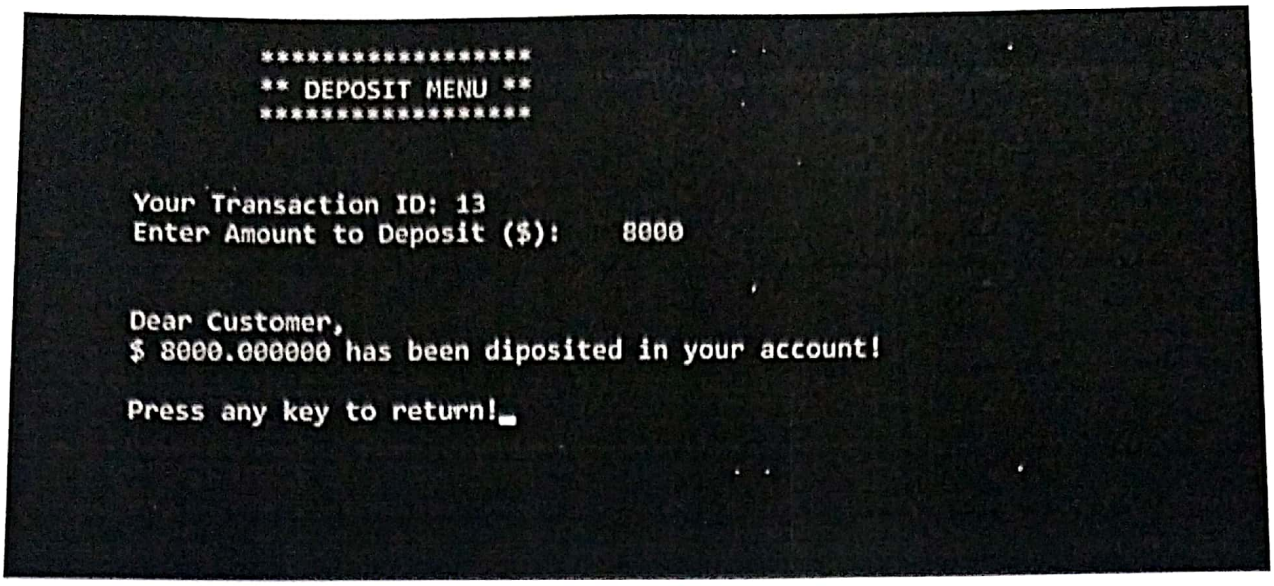


Figure 8: Deposit Menu Sample output

The customer can deposit desired amount of money as shown in the figure above. They are asked to enter the amount to deposit and after entering the desired amount to deposit, the amount is deposited to their account number.

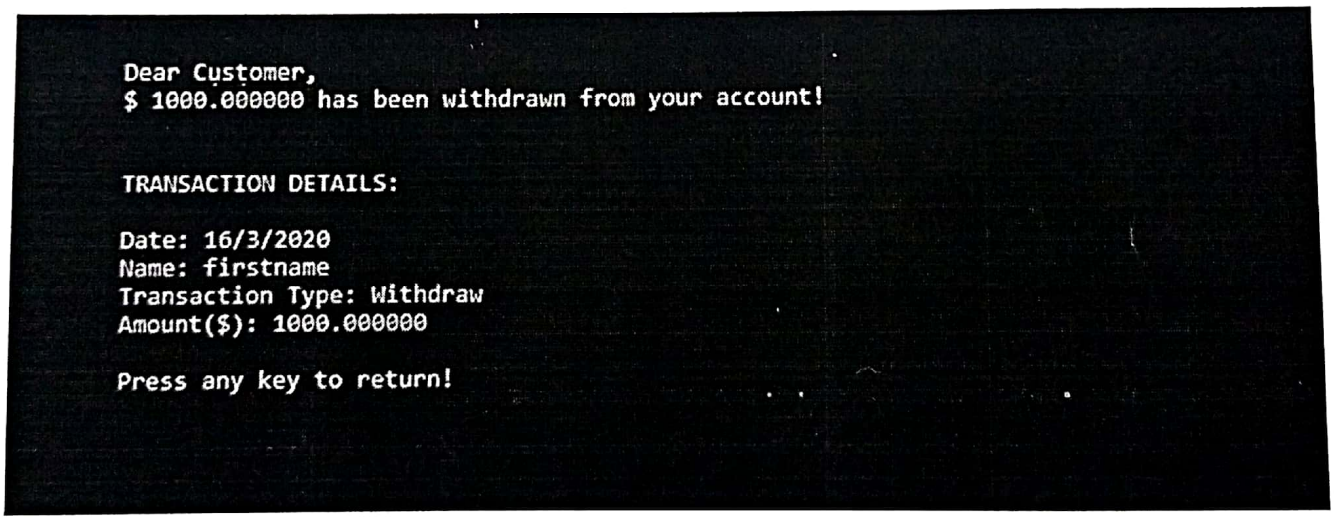
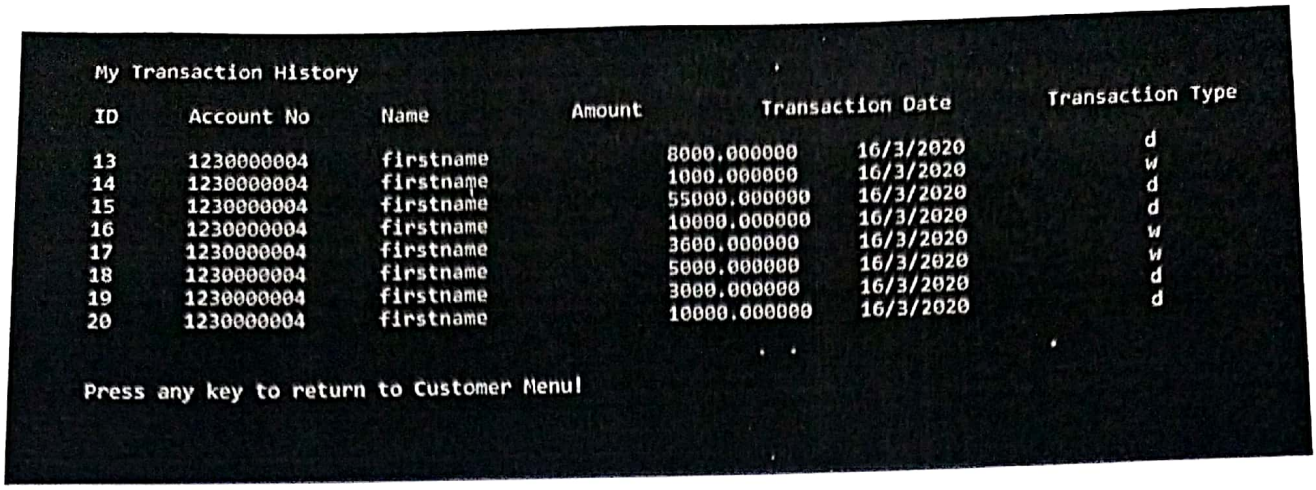


Figure 9: Withdraw Sample output

In withdraw menu when a user inputs certain amount to withdraw, the above output is shown. In the above figure, the user entered $1000 withdraw and the output is shown.



My Transaction History

ID	Account No	Name	Amount	Transaction Date	Transaction Type
13	1230000004	firstname	8000.000000	16/3/2020	d
14	1230000004	firstname	1000.000000	16/3/2020	w
15	1230000004	firstname	55000.000000	16/3/2020	d
16	1230000004	firstname	10000.000000	16/3/2020	d
17	1230000004	firstname	3600.000000	16/3/2020	w
18	1230000004	firstname	5000.000000	16/3/2020	w
19	1230000004	firstname	3000.000000	16/3/2020	d
20	1230000004	firstname	10000.000000	16/3/2020	d

Press any key to return to Customer Menu!

Figure 10: Transaction History Sample output

After performing any type of transaction, the transaction record is stored in file and there is separate function to access the transaction history of a customer. The above figure shows the output when user chooses to see the transaction history from customer menu.

*Screenshort seems quiet-good.

8. Conclusion

MUBMS is a console based non graphical application compiled using C-Programming language. The application was developed with the aim to provide quality mobile banking experience to the customers and a database management platform for the bank. With the use of this application, the administrators can increase their productivity with a lot of work done in few seconds of time. The easy, smooth and fast operation of the app will definitely serve customer in reliable manner. With the database management of the customers, the bank can perform smoothly without being engaged in database paperwork and filing.

The development of this mini project helped me in depth understanding of the C-Programming concepts like iterative constructs, arrays, structures, conditional statements. Moreover, it helped me to develop logical understanding and thinking.

9. References

Tutorial point, 2018. C_header file. [Online] Available at: https://www.tutorialspoint.com/cprogramming/c_header_files.htm [Accessed 22 02 2020].

Learn-c.org. (2019). *Learn C - Free Interactive C Tutorial*. [online] Available at: https://www.learn-c.org/.

www.programiz.com. (n.d.). *C struct (Structures)*. [online] Available at: https://www.programiz.com/c-programming/c-structures.

* Conclusion and references should be in different pages.

* Very few references. Listmore.