

# DEPARTMENT OF COMPUTER SCIENCE AND APPLICATIONS

**PANJAB UNIVERSITY**

**CHANDIGARH**



**C project On**

**“Bank Management System”**

**SUBMITTED TO:**

**Prof. M Syamala Devi**

**SUBMITTED BY:**

**SRISHTI**

**MCA –**

**1(Morning) Roll**

**No - 39**

## **CERTIFICATE**

This is to certify that **Srishti (Roll no. 39)**, student of MCA-1 (Morning) has successfully completed his project “**Bank Management System**” under my guidance positively. The project progress has been continuously reported and has been in my knowledge consistently. The project is the original work done by the mentioned student only.

**Date:**

**Project Guide:**

Prof. M Syamala Devi

## **ACKNOWLEDGEMENT**

A formal statement of acknowledgement is hardly sufficient to express our gratitude towards the personalities who have helped us to undertake and carry out this project. I hereby convey our thankfulness and obligation to my parents, friends and my professor who provided valuable help, support and guidance to carry on this project. I take this opportunity to express my deepest gratitude to those who have generously helped me in providing the valuable knowledge and expertise during my Project.

I am thankful to **Prof. M Syamala Devi** professor of Department of Computer Science and Applications (DCSA), Panjab University, Chandigarh, for her thorough guidance right from first day till the end of the Project and giving me the required guidance and removing any difficulties faced by me during the project.

I would like to thank each and every person who has contributed in any of the ways in my training. I thank the almighty god for blessing me with new challenges in life and giving me enough strength to meet those challenges.

# Table of Contents

<b>INTRODUCTION TO PROJECT TITLE .....</b>	<b>5</b>
<b>AIM AND OBJECTIVES .....</b>	<b>6-7</b>
<b>PROJECT DESIGN AND DESCRIPTION.....</b>	<b>8-10</b>
<b>TECHNOLOGY USED AND IMPLEMENTATION.....</b>	<b>11-14</b>
<b>SOURCE CODE.....</b>	<b>15-44</b>
<b>OUTPUT .....</b>	<b>45-50</b>
<b>TESTING WITH VALIDATIONS.....</b>	<b>51-52</b>
<b>SIGNIFICANCE OF THE PROJECT.....</b>	<b>53-54</b>
<b>BIBLIOGRAPHY/ REFERENCES.....</b>	<b>55</b>

# **INTRODUCTION**

The bank management system project is a program that keeps track of a client's bank account. This project demonstrates the operation of a banking account system and covers the essential functions of bank management software. It develops a project for resolving a customer's financial applications in a banking environment to meet the needs of an end banking user by providing multiple ways to complete banking chores.

Additionally, this project is to provide additional features to the user's workspace that are not available in a traditional banking project. The project's bank management system is built on cutting-edge technologies. This project's main goal is to create software for a bank account management system. This project was designed to make it simple and quick to complete previously impossible processes with manual systems which are now possible with this software.

The code is written in C language. The project banking management is designed to automatic record procedure making less paper work and automatic the process of calculation of interest.

## **AIM OF THE PROJECT**

Any project without aim would be like “a ship without captain”. It is the AIM that gives project a life. So, it is very important for a project to have some aim and the project should be developed in accordance to serve the aim. The project “Banking Management System” has been developed by keeping various aims in mind. Some of these aims are: -

- The major aim is to target as much customers as possible. The best part of this project is that the target area is quite wide as this project can be used to open and manipulate the account of any age group regardless of the gender of the client.
- Another aim is to get fast and accurate results to the transactions, which is must to avoid economic crisis.
- One more aim is to make easy to use interface which will allow bank employees to operate it easily and efficiently.

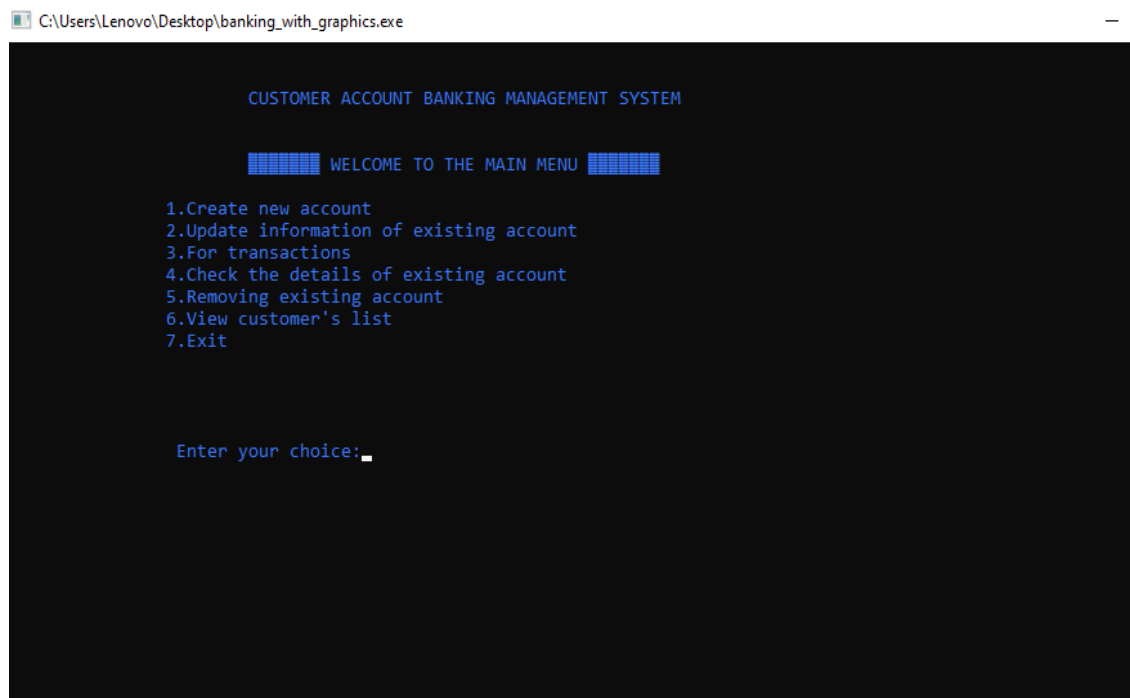
## **PROJECT DETAILS AND OBJECTIVES:**

Now let us start with the detailed description (detailing) of the project. My project “Banking Management System” has a lot of key features which eventually contributes to enhance the popularity of the project. Some of the key features are discussed as follows:

- All the validations have been implemented to ensure the correctness of the data supplied.
- As far as technicalities are concerned, the code has been properly commented so that the code is readable.
- The project has been developed by taking care of the time complexity of the project, so that the results are faster.
- File manipulations have been used to store the data permanently and manipulate that data later on. Along with these features, there are some objectives of the projective.

These objectives are must to achieve the proper and efficient functionality of the project. All these objectives have been achieved in this project. Some of these objectives are as follows:

- To create new account of the person with correct personal details. And to delete an existing account.
- To view all transactions related to a particular account.
- To efficiently withdraw the amount from a particular bank account.
- To efficiently deposit a particular amount to a particular bank account.
- To modify and delete any account with ease. To search the details of account by account number or by name.
- To store and retrieve correct data from input and output source.
- To maintain privacy using admin privileged passwords.



```
C:\Users\Lenovo\Desktop\banking_with_graphics.exe

CUSTOMER ACCOUNT BANKING MANAGEMENT SYSTEM

      WELCOME TO THE MAIN MENU

1.Create new account
2.Update information of existing account
3.For transactions
4.Check the details of existing account
5.Removing existing account
6.View customer's list
7.Exit

Enter your choice: _
```

## **PROJECT DESCRIPTION**

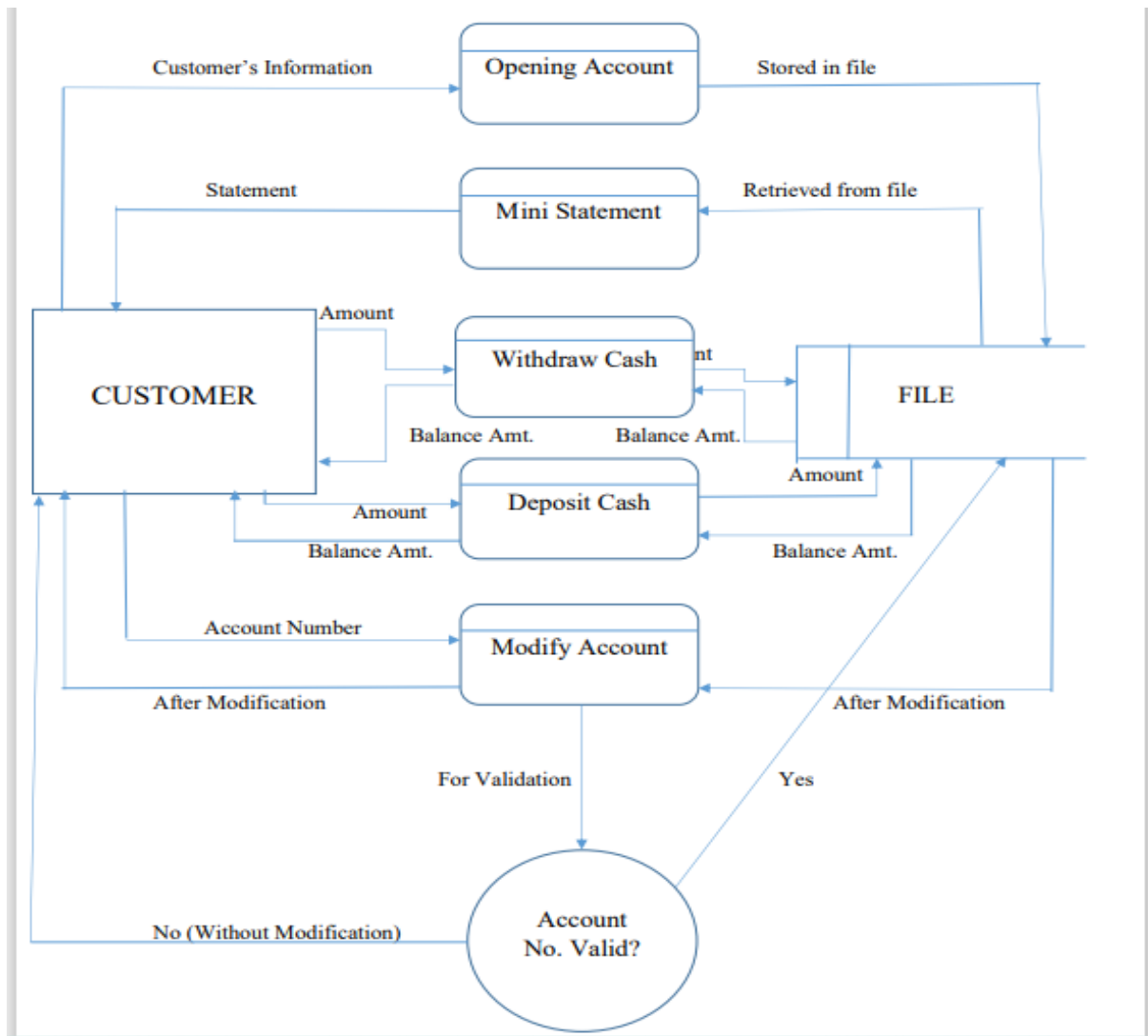
- A menu driven program which provides user with various options of banking features.
- User can create a new account.
- User can delete an existing bank account.
- User can edit a user's information very easily and fast.
- User can know it's interest value according to its bank account type.
- User can deposit and withdraw the given amount of money from one account to another account.



# PROJECT DESIGN

After clearly understanding the requirements of the target audience, the Design phase begins. This involves planning the outlook of the project and laying the blueprint of the layout. It is an important step in the development cycle as it is going target audience.

THE DATA FLOW DIAGRAM IS AS FOLLOWS:



### SWITCH CHOICE1 CONSISTS:

- Create new account which include:
- Today's date, name, citizenship number
- Gender, account number, phone number, type of account

### SWITCH CHOICE2 CONSISTS:

- Update existing account information by taking account number.

### SWITCH CHOICE3 CONSISTS:

- FOR TRANSACTIONS (to add or withdraw some amount of money)

### SWITCH CHOICE4 CONSISTS:

- Gives the details of account by entering account number or by entering name of the account holder.

### SWITCH CHOICE5 CONSISTS:

- To remove the existing account.

### SWITCH CHOICE6 CONSISTS:

- View the customer's list.

## TECHNOLOGY USED:



**CLanguage:** C programming is a general-purpose, procedural, imperative computer programming language developed in 1972 by Dennis M. Ritchie at the Bell Telephone Laboratories to develop the UNIX operating system. C is the most widely used computer language. It keeps fluctuating at number one scale of popularity along with Java programming language, which is also equally popular and most widely used among modern software programmers.

## FEATURES OF C USED:

### Header Files included:

**#include<stdio.h>** -stdio.h is a header file which has the necessary information to include the input/output related functions in our program. Example printf ,scanf etc.

**#include<stdlib.h>**- The stdlib.h header defines four variable types, several macros, and various functions for performing general functions.

**#include<string.h>**- The **string.h** header defines one variable type, one macro, and various functions for manipulating arrays of characters.

**#include<windows.h>**- **windows.h** is a Windows-specific header file for the C and C++ programming languages which contains declarations for all of the functions in the Windows API, all the common macros used by Windows programmers, and all the data types used by the various functions and subsystems. It defines a very large number of Windows specific functions that can be used in C.

**Conditional Statements:** Conditional statements execute sequentially when there is no condition around the statements. If you put some condition for a block of statements, the execution flow may change based on the result evaluated by the condition. This process is called decision making in 'C.' In 'C' programming conditional statements are possible with the help of the following two constructs:

If statement

If-else statement

**Loops** :A loop statement allows us to execute a statement or group of statements multiple times .Loops used the project are:

For loop

While loop

do...while loop

**Loop control statements used:**

Goto statement

Continue statement

Break statement

## **DECISION CONTROL STATEMENTS:**

In any programming language, there is a need to perform different tasks based on the condition.

Using decision control statements, we can control the flow of program in such a way so that it executes certain statements based on the outcome of a condition (i.e. true or false).

Decision control statements used are

Switch case statements.

**ErrorControlsUsed:**As such ,C programming does not provide direct support for error handling but being a system programming language, it provides you access at lower level in the form of return values.

Error controls have been applied in the main menu and its every sub menu very carefully and precisely. Its been taken care of that the program doesn't exit or stops responding if any wrong input is entered.

A error message is displayed and the screen waits for the user to input correct options.

**File System Used:**

Creating a file, Read from file ,Write in a file

# IMPLEMENTATION:

After figuring what needs to be done, the resources needed to conduct the project are analyzed and utilized. Towards the end of this phase, the system and software requirements were stated. These included:

## SYSTEM REQUIREMENTS

### **Software and editor used:**

Source Code language: C (Developed By dennisritchie)

OS : Windows 10 Pro 64-bit

Compiler DEV C++

### **3.1 Minimum System Requirements:**

Software Requirements :

. Windows 7, XP, 10, 11

.Windows 64 bit or 32 bit

.Mac OS 10.4 and above

### **3.2 Minimum Hardware Requirements :**

.Intel Pentium 4 and above

.10 GB hard disk or above.

.Minimum Ram 512MB or above

## **SOURCE CODE**

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
#include<windows.h>
int i,j;
int main_exit;
void menu();
struct date{
    int month,day,year;

};
struct {

    char name[60];
    int acc_no;
    int age;
    char address[60];
    char citizenship[15];
    char phone[12];
    char acc_type[10];
    float amt;
    struct date dob;
```

```
struct date deposit;
```

```
struct date withdraw;
```

```
}add,upd,check,rem,transaction;
```

```
float interest(float t,float amount,int rate)// function to calculate the  
interest on loan
```

```
{
```

```
float SI;
```

```
SI=(rate*t*amount)/100.0;
```

```
return (SI);
```

```
}
```

```
void fordelay(int j)
```

```
{ int i,k;
```

```
for(i=0;i<j;i++)
```

```
    k=i;
```

```
}
```

```
void new_acc() //creating a new account
```

```
{
```

```
int choice;
```

```
FILE *ptr;
```



```
ptr=fopen("record.txt","a+");
```

```
account_no:
```

```
    one:
```

```
system("cls");
```

```
printf("\t\t\t\t ADD RECORD ");
```

```
printf("\n\n\nEnter today's date(mm/dd/yyyy):");
```

```
scanf("%d/%d/%d",&add.deposit.month,&add.deposit.day,&add.deposit.y  
ear);
```

```
printf("\n\nEnter the account number:");
```

```
scanf("%d",&check.acc_no);
```

```
/* if(strlen(check.acc_no)!=7)
```

```
{
```

```
    printf(" 7 digit required!! type again");
```

```
    fordelay(1000000000);
```

```
    goto account_no;
```

```
}*/
```

```
while(fscanf(ptr,"%d %s %d/%d/%d %d %s %s %s %s %f  
%d/%d/%d\n",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&a  
dd.dob.year,&add.age,add.address,add.citizenship,add.phone,add.acc_type
```

```
,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=E
OF)
{
    if (check.acc_no==add.acc_no)
        {printf("Account no. already in use!");
        fordelay(1000000000);
        goto account_no;

        }
}
add.acc_no=check.acc_no;
printf("\nEnter the name:");
scanf("%s",&add.name);
printf("\nEnter the date of birth(mm/dd/yyyy):");
scanf("%d/%d/%d",&add.dob.month,&add.dob.day,&add.dob.year);
printf("\nEnter the age:");
scanf("%d",&add.age);
if((add.age)<18)// error control
{
    printf("you are not eligible for opening an account");
    fordelay(1000000000);
    goto account_no;
}
printf("\nEnter the address:");
```

```
scanf("%s",&add.address);  
printf("\nEnter the citizenship number:");  
scanf("%s",&add.citizenship);  
if(strlen(add.citizenship)!=10)//error control  
{  
    printf("citizenship number should be of 10 digit ");  
    fordelay(1000000000);  
    goto account_no;  
  
}
```

```
printf("\nEnter the phone number: ");  
scanf("%s",&add.phone);  
if(strlen(add.phone)!=10)//error control  
{  
    printf("Phone number should be of 10 digit ");  
    fordelay(1000000000);  
    goto account_no;  
  
}
```

```
printf("\nEnter the amount to deposit:$");  
scanf("%f",&add.amt);
```

```
printf("\nType of account:\n\t#Saving\n\t#Current\n\t#Fixed1(for 1
year)\n\t#Fixed2(for 2 years)\n\t#Fixed3(for 3 years)\n\n\tEnter your
choice:");
```

```
scanf("%s",&add.acc_type);
```

```
fprintf(ptr,"%d %s %d/%d/%d %d %s %s %s %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);
```

```
fclose(ptr);
```

```
printf("\nAccount created successfully!");
```

```
add_invalid:
```

```
printf("\n\n\n\t\tEnter 1 to go to the main menu and 0 to exit:");
```

```
scanf("%d",&main_exit);
```

```
system("cls");
```

```
if (main_exit==1)
```

```
    menu();
```

```
else if(main_exit==0)
```

```
    close();
```

```
else
```

```
{
```

```
    printf("\nInvalid!\a");
```

```
    goto add_invalid;
```

```
}
```

```
}
```

```
void view_list()//to view all customer's list
```

```
{
```

```
FILE *view;
```

```
view=fopen("record.txt","r");
```

```
int test=0;
```

```
system("cls");
```

```
printf("\nACC. NO.\tNAME\t\tADDRESS\t\tPHONE\n");
```

```
while(fscanf(view,"%d %s %d/%d/%d %d %s %s %s %s %f  
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&ad  
d.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc_typ  
e,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=E  
OF)
```

```
{
```

```
printf("\n%6d\t  
%10s\t\t\t%10s\t\t\t%s",add.acc_no,add.name,add.address,add.phone);
```

```
test++;
```

```
}
```

```
fclose(view);
```

```
if (test==0)
```

```
{ system("cls");
```

```
printf("\nNO RECORDS!!\n");}
```

```
view_list_invalid:
```

```
printf("\n\nEnter 1 to go to the main menu and 0 to exit:");
```

```
scanf("%d",&main_exit);
```

```
system("cls");
```

```
if (main_exit==1)
```

```
    menu();
```

```
else if(main_exit==0)
```

```
    close();
```

```
else
```

```
{
```

```
    printf("\nInvalid!\a");
```

```
    goto view_list_invalid;
```

```
}
```

```
}
```

```
void edit(void)//to edit the record of existing account
```

```
{
```

```
    int choice,test=0;
```

```
    FILE *old,*newrec;
```

```
    old=fopen("record.txt","r");
```

```
    newrec=fopen("new.txt","w");
```

```

printf("\nEnter the account no. of the customer whose info you want to
change:");

scanf("%d",&upd.acc_no);

while(fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&ad
d.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc_typ
e,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=E
OF)

{
    if (add.acc_no==upd.acc_no)
    {   test=1;

        printf("\nWhich information do you want to
change?\n1.Address\n2.Phone\n\nEnter your choice(1 for address and 2
for phone):");

        scanf("%d",&choice);

        system("cls");

        if(choice==1)

            {printf("Enter the new address:");

                scanf("%s",upd.address);

                fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,upd.address,add.citizenship,add.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);

                    system("cls");

                    printf("Changes saved!");

            }
    }
}

```

```

else if(choice==2)
{
    printf("Enter the new phone number:");
    scanf("%lf",&upd.phone);

    fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,add.address,add.citizenship,upd.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);

    system("cls");
    printf("Changes saved!");
}

}

else

    fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);

}

fclose(old);
fclose(newrec);
remove("record.txt");
rename("new.txt","record.txt");

if(test!=1)

```



```

    {   system("cls");
        printf("\nRecord not found!!\a\a\a");
        edit_invalid:  //labels used for goto statement
        printf("\nEnter 0 to try again,1 to return to main menu and 2 to
exit:");
        scanf("%d",&main_exit);
        system("cls");
        if (main_exit==1)

            menu();
        else if (main_exit==2)
            close();
        else if(main_exit==0)
            edit();
        else
            {printf("\nInvalid!\a");
              goto edit_invalid;}
    }
else
    {printf("\n\nEnter 1 to go to the main menu and 0 to exit:");
      scanf("%d",&main_exit);
      system("cls");
      if (main_exit==1)
          menu();
    }

```

```

else
    close();
}
}

void transact(void)//to add or withdraw monet
{
    int choice,test=0;
    FILE *old,*newrec;
    old=fopen("record.txt","r");
    newrec=fopen("new.txt","w");

    printf("Enter the account no. of the customer:");
    scanf("%d",&transaction.acc_no);

    while (fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&ad
d.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc_typ
e,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=E
OF)
    {

        if(add.acc_no==transaction.acc_no)
        {
            test=1;

            if(strcmpi(add.acc_type,"fixed1")==0||strcmpi(add.acc_type,"fixed2")==0|
|strcmpi(add.acc_type,"fixed3")==0)

```

```

{
    printf("\a\a\a\n\nYOU CANNOT DEPOSIT OR
WITHDRAW CASH IN FIXED ACCOUNTS!!!!");
    fordelay(1000000000);
    system("cls");
    menu();

}

printf("\n\nDo you want to\n1.Deposit\n2.Withdraw?\n\nEnter
your choice(1 for deposit and 2 for withdraw:");
scanf("%d",&choice);
if (choice==1)
{
    printf("Enter the amount you want to deposit:$ ");
    scanf("%f",&transaction.amt);
    add.amt+=transaction.amt;

    fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);

    printf("\n\nDeposited successfully!");
}
else
{
    printf("Enter the amount you want to withdraw:$ ");

```

```

        scanf("%f",&transaction.amt);

        add.amt-=transaction.amt;

        fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);

        printf("\n\nWithdrawn successfully!");
    }

}

else
{

    fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);

}

}

fclose(old);
fclose(newrec);
remove("record.txt");
rename("new.txt","record.txt");
if(test!=1)
{

    printf("\n\nRecord not found!!");
}

```

```
    transact_invalid:
    printf("\n\n\nEnter 0 to try again,1 to return to main menu and 2 to
exit:");
    scanf("%d",&main_exit);
    system("cls");
    if (main_exit==0)
        transact();
    else if (main_exit==1)
        menu();
    else if (main_exit==2)
        close();
    else
    {
        printf("\nInvalid!");
        goto transact_invalid;
    }

}

else
{
    printf("\nEnter 1 to go to the main menu and 0 to exit:");
    scanf("%d",&main_exit);
    system("cls");
    if (main_exit==1)
```

```

        menu();
    else
        close();
}

}

void erase(void)// to delete an existing account
{
    FILE *old,*newrec;
    int test=0;
    old=fopen("record.txt","r");
    newrec=fopen("new.txt","w");
    printf("Enter the account no. of the customer you want to delete:");
    scanf("%d",&rem.acc_no);

    while (fscanf(old,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&ad
d.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc_typ
e,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=E
OF)
    {
        if(add.acc_no!=rem.acc_no)

            fprintf(newrec,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.
year,add.age,add.address,add.citizenship,add.phone,add.acc_type,add.amt,
add.deposit.month,add.deposit.day,add.deposit.year);
    }
}

```

```
else
    {test++;
    printf("\nRecord deleted successfully!\n");
    }
}
fclose(old);
fclose(newrec);
remove("record.txt");
rename("new.txt","record.txt");
if(test==0)
{
    printf("\nRecord not found!!\a\a\a");
    erase_invalid:
    printf("\nEnter 0 to try again,1 to return to main menu and 2 to
exit:");
    scanf("%d",&main_exit);

    if (main_exit==1)
        menu();
    else if (main_exit==2)
        close();
    else if(main_exit==0)
        erase();
```

```
        else
            {printf("\nInvalid!\a");
              goto erase_invalid;}
    }
else
    {printf("\nEnter 1 to go to the main menu and 0 to exit:");
      scanf("%d",&main_exit);
      system("cls");
      if (main_exit==1)
          menu();
      else
          close();
    }
}
```

```
void see(void)
{
    FILE *ptr;
    int test=0,rate;
    int choice;
    float time;
    float intrst;
    ptr=fopen("record.txt","r");
```



```
printf("Do you want to check by\n1.Account no\n2.Name\nEnter your choice:");
```

```
scanf("%d",&choice);
```

```
if (choice==1)
```

```
{ printf("Enter the account number:");
```

```
scanf("%d",&check.acc_no);
```

```
while (fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f %d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&add.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
```

```
{
```

```
if(add.acc_no==check.acc_no) { system("cls");
```

```
test=1;
```

```
printf("\nAccount NO.:%d\nName:%s \nDOB:%d/%d/%d\nAge:%d \nAddress:%s \nCitizenship No:%s \nPhone number:%.0lf\nType Of Account:%s \nAmount deposited:$ %.2f \nDate Of Deposit:%d/%d/%d\n\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,
```

```
add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);
```

```
if(strcmpi(add.acc_type,"fixed1")==0)
```

```
{
```

```
        time=1.0;
        rate=9;
        intrst=interest(time,add.amt,rate);
        printf("\n\nYou will get $%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+1);
    }
    else if(strcmpi(add.acc_type,"fixed2")==0)
    {
        time=2.0;
        rate=11;
        intrst=interest(time,add.amt,rate);
        printf("\n\nYou will get $%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+2);

    }
    else if(strcmpi(add.acc_type,"fixed3")==0)
    {
        time=3.0;
        rate=13;
        intrst=interest(time,add.amt,rate);
        printf("\n\nYou will get $%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+3);

    }
```

```

        else if(strcmpi(add.acc_type,"saving")==0)
        {
            time=(1.0/12.0);
            rate=8;
            intrst=interest(time,add.amt,rate);
            printf("\n\nYou will get $.%.2f as interest on %d of every
month",intrst,add.deposit.day);

        }
        else if(strcmpi(add.acc_type,"current")==0)
        {

            printf("\n\nYou will get no interest\n");

        }

    }
}
else if (choice==2)
{
    printf("Enter the name:");
    scanf("%s",&check.name);

    while (fscanf(ptr,"%d %s %d/%d/%d %d %s %s %lf %s %f
%d/%d/%d",&add.acc_no,add.name,&add.dob.month,&add.dob.day,&ad

```

```
d.dob.year,&add.age,add.address,add.citizenship,&add.phone,add.acc_type,&add.amt,&add.deposit.month,&add.deposit.day,&add.deposit.year)!=EOF)
```

```
{
```

```
    if(strcmpi(add.name,check.name)==0)
```

```
    {    system("cls");
```

```
        test=1;
```

```
        printf("\nAccount No.:%d\nName:%s \nDOB:%d/%d/%d\nAge:%d \nAddress:%s \nCitizenship No:%s \nPhone number:%.0lf\nType Of Account:%s \nAmount deposited:$%.2f \nDate Of Deposit:%d/%d/%d\n\n",add.acc_no,add.name,add.dob.month,add.dob.day,add.dob.year,add.age,add.address,add.citizenship,add.phone,
```

```
add.acc_type,add.amt,add.deposit.month,add.deposit.day,add.deposit.year);
```

```
        if(strcmpi(add.acc_type,"fixed1")==0)
```

```
        {
```

```
            time=1.0;
```

```
            rate=9;
```

```
            intrst=interest(time,add.amt,rate);
```

```
            printf("\n\nYou will get $%.2f as interest on %d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+1);
```

```
        }
```

```
        else if(strcmpi(add.acc_type,"fixed2")==0)
```

```
        {
```

```
            time=2.0;
```

```

        rate=11;

        intrst=interest(time,add.amt,rate);

        printf("\n\nYou will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+2);

    }

    else if(strcmpi(add.acc_type,"fixed3")==0)
    {
        time=3.0;
        rate=13;
        intrst=interest(time,add.amt,rate);
        printf("\n\nYou will get $.%.2f as interest on
%d/%d/%d",intrst,add.deposit.month,add.deposit.day,add.deposit.year+3);

    }

    else if(strcmpi(add.acc_type,"saving")==0)
    {
        time=(1.0/12.0);
        rate=8;
        intrst=interest(time,add.amt,rate);
        printf("\n\nYou will get $.%.2f as interest on %d of every
month",intrst,add.deposit.day);

    }

```

```
        else if(strcmpi(add.acc_type,"current")==0)
        {

                printf("\n\nYou will get no interest\n");

        }

    }
}
}
```

```
fclose(ptr);
if(test!=1)
{
    system("cls");
    printf("\nRecord not found!!\n");
    see_invalid:
    printf("\nEnter 0 to try again,1 to return to main menu and 2 to
exit:");
    scanf("%d",&main_exit);
    system("cls");
    if (main_exit==1)
        menu();
}
```

```
        else if (main_exit==2)
            close();
        else if(main_exit==0)
            see();
        else
        {
            system("cls");
            printf("\nInvalid!\a");
            goto see_invalid;}
    }
else
    {printf("\nEnter 1 to go to the main menu and 0 to exit:");
    scanf("%d",&main_exit);}
    if (main_exit==1)
    {
        system("cls");
        menu();
    }

    else
    {

        system("cls");
        close();
```

```
}
```

```
}
```

```
void close(void)
```

```
{
```

```
    printf("\n\n\nThis C Project is developed by srishti");
```

```
}
```

```
void menu(void)
```

```
{    int choice;
```

```
    system("cls");
```

```
    system("color 9");
```

```
    printf("\n\n\t\tCUSTOMER ACCOUNT BANKING  
MANAGEMENT SYSTEM");
```

```
    printf("\n\n\n\t\t\xB2\xB2\xB2\xB2\xB2\xB2\xB2 WELCOME TO  
THE MAIN MENU \xB2\xB2\xB2\xB2\xB2\xB2\xB2");
```

```
    printf("\n\n\t\t1.Create new account\n\t\t2.Update information of  
existing account\n\t\t3.For transactions\n\t\t4.Check the details of existing  
account\n\t\t5.Removing existing account\n\t\t6.View customer's  
list\n\t\t7.Exit\n\n\n\n\t\tEnter your choice:");
```

```
    scanf("%d",&choice);
```



```
system("cls");  
switch(choice)  
{  
    case 1:new_acc();  
    break;  
    case 2:edit();  
    break;  
    case 3:transact();  
    break;  
    case 4:see();  
    break;  
    case 5:erase();  
    break;  
    case 6:view_list();  
    break;  
    case 7:close();  
    break;  
}
```

```
}  
int main()
```

```
{  
    char pass[10],password[10]="admin";  
    int i=0;  
    printf("\n\n\t\tEnter the password to login:");  
    scanf("%s",&pass);  
    /*do  
    {  
    //if (pass[i]!=13&&pass[i]!=8)  
        {  
            printf("*");  
            pass[i]=getch();  
            i++;  
        }  
    }while (pass[i]!=13);  
    pass[10]='\0';*/  
    if (strcmp(pass,password)==0)  
        {printf("\n\nPassword Match!\nLOADING");  
        for(i=0;i<=4;i++)  
        {  
            fordelay(1000000000);  
            printf(".");  
        }  
        system("cls");  
        menu();  
    }
```

```
    }  
else  
    {   printf("\n\nWrong password!!\a\a\a");  
        login_try:  
        printf("\nEnter 1 to try again and 0 to exit:");  
        scanf("%d",&main_exit);  
        if (main_exit==1)  
            {  
  
                system("cls");  
                main();  
            }  
  
        else if (main_exit==0)  
            {  
                system("cls");  
                close();}  
        else  
            {printf("\nInvalid!");  
              fordelay(1000000000);  
              system("cls");  
              goto login_try;}  
    }
```

```
return 0;  
}
```

## TESTING:

After the completion of the coding process, the testing phase comes. This phase is responsible to check for bugs and errors. And also put the project into testing for a limited group of people to understand the changes that can be made.

The source code declared above for the program of the Banking management system has been tested and it has been found that above code is correct, the program involves many features. These operations are done very carefully.

There are two types of testing involved:

System Testing

Integration Testing

System Testing involves whole testing of the program at once and

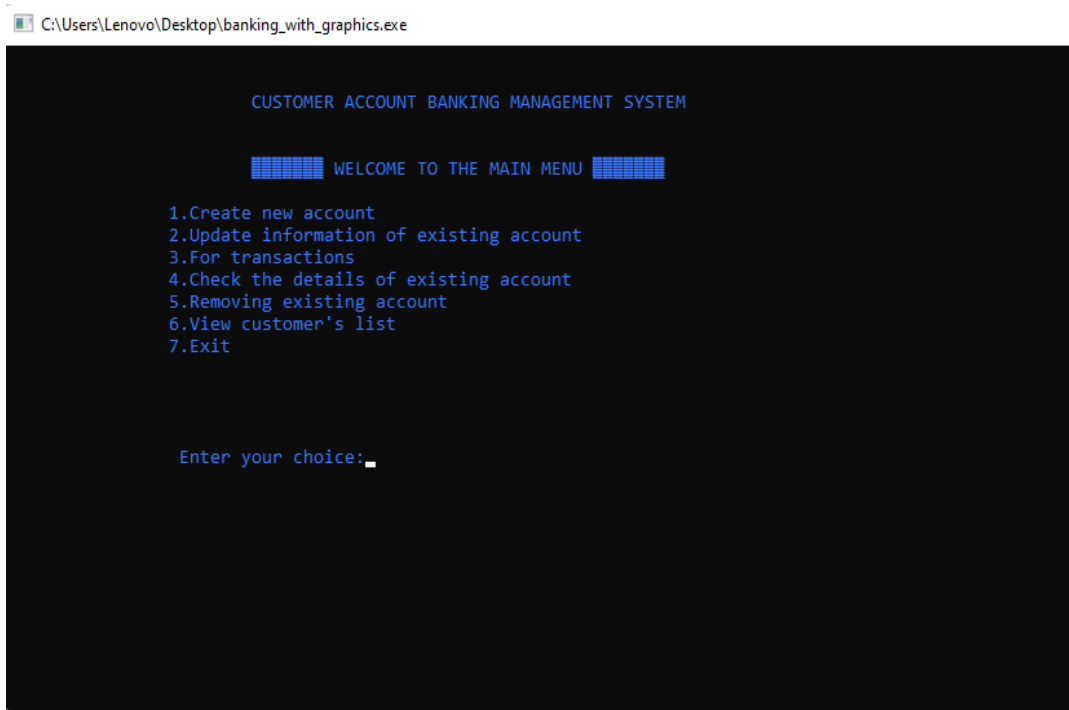
Integration Testing involves the breaking of program to modules and then test.

**Ms. M. Syamala Devi** was involved in this phase. She suggested to make some changes in the project layout. All in all, the necessary changes were made after testing the minor project.

# OUTPUT

TO LOGIN USE PASSWORD=" admin"

MAIN MENU:



```
C:\Users\Lenovo\Desktop\banking_with_graphics.exe

CUSTOMER ACCOUNT BANKING MANAGEMENT SYSTEM

##### WELCOME TO THE MAIN MENU #####

1.Create new account
2.Update information of existing account
3.For transactions
4.Check the details of existing account
5.Removing existing account
6.View customer's list
7.Exit

Enter your choice: _
```

1.Create new account:

```
C:\Users\Lenovo\Desktop\banking_with_graphics.exe
ADD RECORD

Enter today's date(mm/dd/yyyy):9/6/2022
Enter the account number:12345678
Enter the name:lata
Enter the date of birth(mm/dd/yyyy):3/11/2001
Enter the age:20
Enter the address:gh5
Enter the citizenship number:2333356750
Enter the phone number: 9434632196
Enter the amount to deposit:$700

Type of account:
#Saving
#Current
#Fixed1(for 1 year)
#Fixed2(for 2 years)
#Fixed3(for 3 years)

Enter your choice:2

Account created successfully!

Enter 1 to go to the main menu and 0 to exit:
```

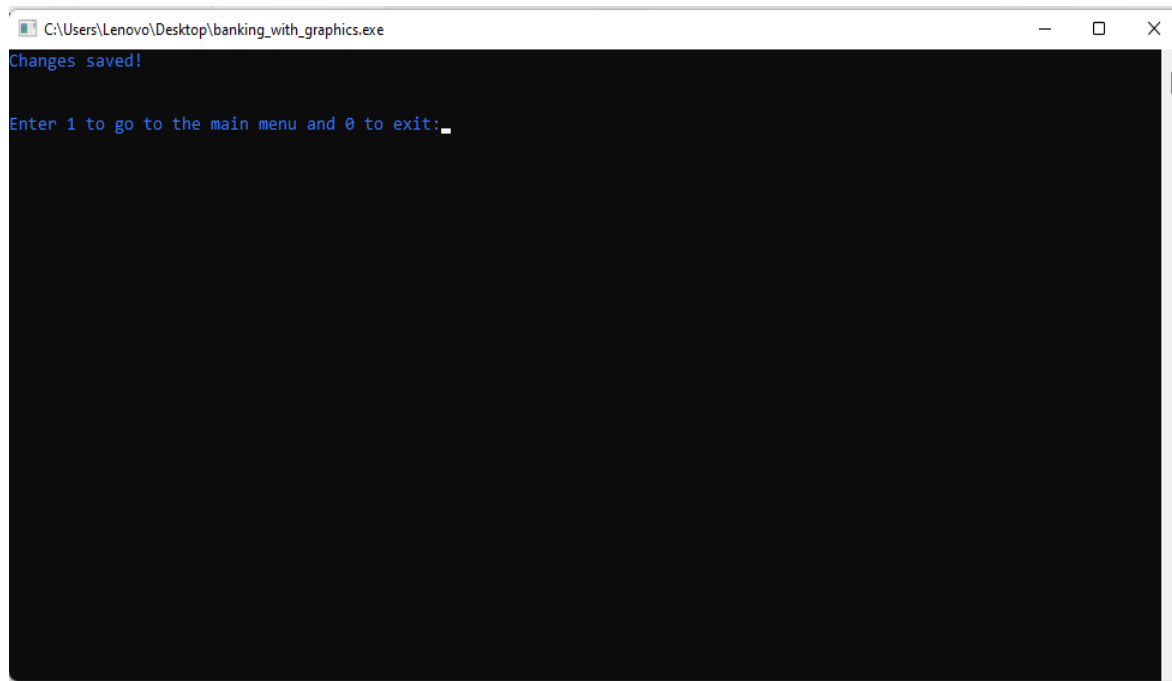
## 2.) Update information of an existing account:

```
C:\Users\Lenovo\Desktop\banking_with_graphics.exe

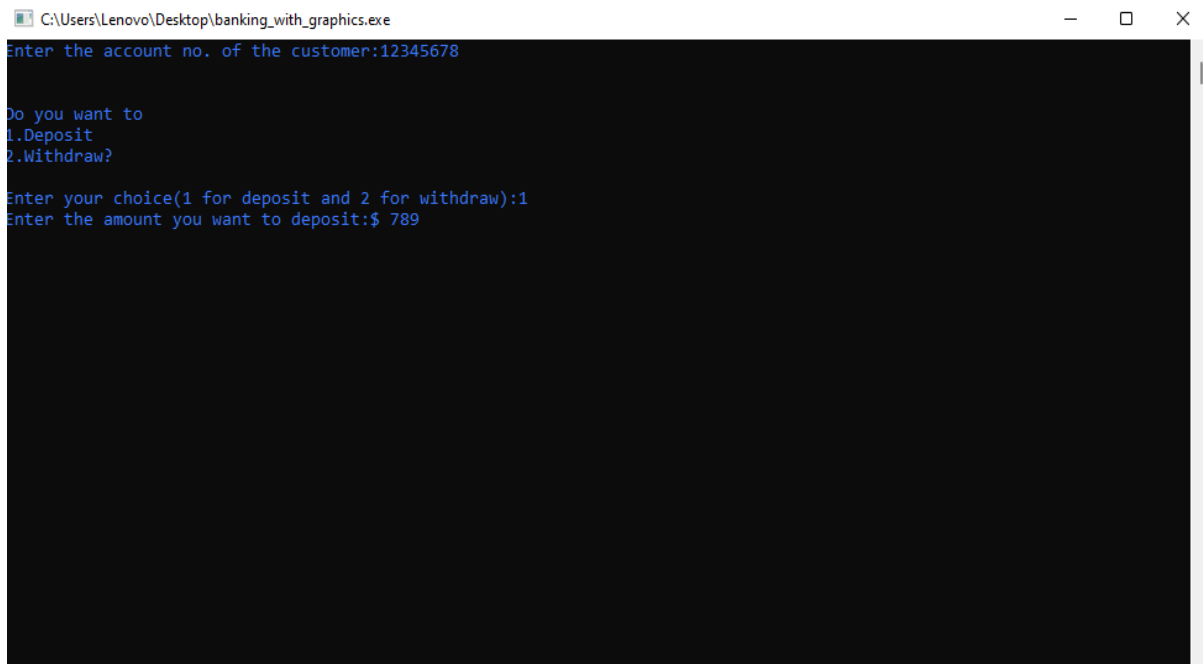
Enter the account no. of the customer whose info you want to change:12345678

Which information do you want to change?
1.Address
2.Phone

Enter your choice(1 for address and 2 for phone):2
```



### 3.) For Transactions:



#### 4.) Check the details of existing account:

```
C:\Users\Lenovo\Desktop\banking_with_graphics.exe
Do you want to check by
1.Account no
2.Name
Enter your choice:2
Enter the name:lata
```

```
C:\Users\Lenovo\Desktop\banking_with_graphics.exe
Account No.:12345678
Name:lata
DOB:3/11/2001
Age:20
Address:gh4
Citizenship No:2333356750
Phone number:0
Type Of Account:2
Amount deposited:$1489.00
Date Of Deposit:4/6/2022

Enter 1 to go to the main menu and 0 to exit:
```

#### 5.) Remove an existing account:



```
C:\Users\Lenovo\Documents\projects\banking_with_graphics.exe
Enter the account no. of the customer you want to delete:12345678
Record deleted successfully!
Enter 1 to go to the main menu and 0 to exit:
```

## 6.) View customer list:

```
C:\Users\Lenovo\Documents\projects\banking_with_graphics.exe

ACC. NO.      NAME      ADDRESS PHONE
12345678      srishti   gh4      9996640785
2345678      smriti    gh4      9876537297
786526717     lata      gh5      8976565389

Enter 1 to go to the main menu and 0 to exit: _
```

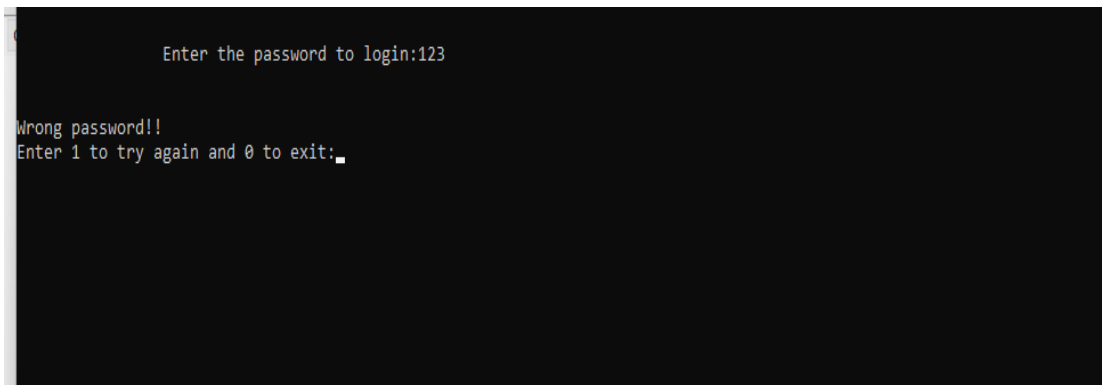
To recapitulate, these were the important options that where been part of the project “Banking Management System”. The code has been included in this report in the form of some important subroutines. Before this implementation code few algorithms were developed and this was possibly the best algorithm out of them. I have tried to make the time complexity less as much as possible. During this implementation phase, the syntax was not an issue, this is because of the convenience of the simple syntax of C which has been studied in the class perfectly. A little problem arose when the in-built functions were having a wide range which were unknown to me being a beginner. But this problem was sorted by proper R&D (research and development). The line of code was another issue, this code is of about 600 lines, which has been quite difficult to handle. This issue has been considered as the limitation of C that the projects developed in this language has lengthy codes. Rest of the thing went pretty well. The graphics worked pretty fine and makes the project quite interesting. The files were handled easily and efficiently. So, implementation phase was full of thrills and interests as far as the “Banking Management System was concerned”.

# **TESTING WITH VALIDATIONS**

Software testing is defined as an activity to check whether the actual results match the expected results and to ensure that the software system is Defect free. It involves execution of a software component or system component to evaluate one or more properties of interest. Software testing also helps to identify errors, gaps or missing requirements in contrary to the actual requirements. It can be either done manually or using automated tools. Some prefer saying Software testing as a White Box and Black Box Testing. In simple terms, Software Testing means Verification of Application Under Test (AUT). For checking the project is running properly under different circumstances following datasets been supplied to the program:

## **• Validation on password:**

The password set for the admin login is “admin”, if any other password is supplied as input then the code has been handled by printing the message “Wrong password” as shown in the following screen shot.



## **• Number of digits in citizenship number:**

The number of digits of an account can only be 10 digits, neither more nor less. If the number of digits in citizenship number is more or less than 10, then the error has been handled by printing the message “citizenship number should be of 10 digit” as shown in the screen shot on the next page.

```
Enter the age:22
Enter the address:gh4
Enter the citizenship number:8987
citizenship number should be of 10 digit
```

- **Validation on age:**

If age is less than 18 then the error has been handled by printing the message “you are not eligible for opening an account” as shown in the screen shot on the next page.

```
Enter the date of birth(mm/dd/yyyy):8/8/2000
Enter the age:9
you are not eligible for opening an account.
```

So these were some of the different datasets that were fed to the program during the testing phase. All the data sets have been handled in this project. This handling of datasets are called validations. Validations are required in every project to ensure the correctness of the data that is being supplied to the program. So overall, the testing phase is one of the important step. Without testing no errors can be removed (if any) from the code. In this project as well, all the validations were applied as the result of the testing done. Before testing, the validations were not applied and the components were not working properly. But it was possible due to testing that all the errors occurred in the program were handled efficiently

# **SIGNIFICANCE OF THE PROJECT**

Now that the project has been developed, it should have some significance. Any project that has been developed would be having some purpose. Significance will tell, how the project would be beneficial for people of different application areas. As far as application areas are concerned, there is a wide range. Out of them, let us select the most popular application areas to understand the significance of the project. The significance of the project will be demonstrated in the following application areas:

- Significance to society
- Significance to IT community
- Significance to Education and Research

**To society:** The project would be having the preponderant significance to the society. The people of the society would be able to open their accounts with their correct personal information. There would be ease for them to deposit or withdraw cash. The faster results of the operations would help them not waiting in the bank queues for long time. The mini statement of their account would be received with proper indentation and clarity. Due to the bank failure, millions of depositors suffered from a great problem, as they didn't get back their deposited money. This problem would no longer exist with the increased security of this application. It is known that customer satisfaction is the most important aspect of the success of the project. Happy faces of the customers matter. The people of society are the customers that would have advantage of this project for their reliable monetary transactions. All in all, the project serves the society in the best possible way.

**To IT community:** The IT community is making the world right now. Information Technology (IT) is hot burning topic in this era. Everything, these days, is being done by using technology by hook or by crook. Every country is focusing on strengthening their IT sector. Although my project is very basic, but still it can be of some significance to IT community. Based on my idea for this project, IT community can think of preparing a bank management system that will be more reliable, secure and faster. Many beginners in the IT field can use platform like GitHub to seek knowledge and many IT professionals can help me improving the project marginally. Whatever you develop there is always a scope. From beginners

to experts everyone can gain and give knowledge at any point of time. Hopefully, my project would bring the interests of many IT experts and beginners, as a result the IT community might benefit from my project and also I can gain some knowledge to improve my skills as well. All in all, the significance to IT community is not assured, but the world is full of possibilities and hopes and I hope I might contribute something to strengthen the IT field.

**To Education and Research:** My project “Banking Management System” has great significance as far as education and research is concerned. Not only IT students but commerce students can also seek a lot from this project. Being myself a student, I know about the problems that arise while researching and being educated. Students and research scholars find it difficult to follow the step to enhance their skills. My project could possibly be the first step towards their goal. The simplicity of this project would make them more interested towards the project and they will try to learn and would also try to implement their knowledge to the project. This way both project and student’s knowledge will be improved. My project might motivate them to think something innovative. So, this way my project can serve the area of Education and Research.

# **BIBLIOGRAPHY/ REFERENCES**

- [C Programming Absolute Beginner's Guide](#) written by Greg Perry and Dean Miller.
- Let Us C by Yashavant Kanetkar.
- Data Structures Using C by Balagurusamy.
- <https://www.tutorialspoint.com/cplusplus/index.htm>
- <https://www.w3schools.com/c/>
- <https://www.geeksforgeeks.org/c-tutorial/>
- <https://beginnersbook.com/2017/08/c-tutorial-for-beginners/>