PROJECT ON BANK MANAGEMENT SYSTEM IN C

**A Project Report for Engineering Project**

###### ***Submitted by***

##### PUSPITA UTHYASANI

***in partial fulfillment for the award of the degree of***

##### **Diploma**

in

Computer science & technology

**Women’s polytechnic,kolkata**

At

**Ardent Computech Pvt. Ltd.**



**Acknowledgement**

I take this opportunity to express my deep gratitude and sincerest thank to my project mentor, miss. Mousita dhar for giving most valuable suggestion, helpful guidance and encouragement in the execution of this project work.

I will like to give a special mention to my colleagues. Last but not the least I am grateful to all the faculty members of Ardent Computech Pvt. Ltd. or their support.

CONTENTS

1. Objectives
2. Introductions
3. Algorithm
4. Sample Code
5. Screen shots
6. Discussion and Program Limitations
7. Summary
8. References

# Objectives

The project is mainly based on following objectives:

* To create a project using C-programming and its features.
* To implement features like control statement, structures and file handlings.
* To be familiar with resource reusability by making user defined function.
* To make the program easy while running it.
* To concise the memory of program as far as possible.
* To get an idea about making a simple project using C.
* To be able to solve problems by Compiling and Debugging.

# Introduction

The project ‘Bank Management’ is the sample project for managing bank accounts. The project aims at developing bank management system using the C language that enables a bank to maintain its customer’s account.

The project demonstrates the creation of a user interface of a system, without the use of C Graphics library.

The application uses basic C function to generate menus, show message boxes and print text on the screen.

The application also implements the concept of structures to define the bank items. It also effectively applies the various C concepts such as file operations, looping and branching constructs and string manipulation functions.

# Algorithm

**Main function**

**Steps**

1. Start with welcome screen
2. Display main menu as below

* 1.add account
* 2.edit account
* 3.delete account
* 4.view info
* 5.close application

1. Get choice from user

Choice:-1 call function new account

Choice:-2 call function edit account

Choice:-3 call function to delete account

Choice:-4 call function view info

Choice:-5 call function to close application

**New Account function**

**Steps**

1.declare file pointer ‘ptr’

2.display categories of account to be added

3.get option from user.Is user want to create new account

Yes:-goto step 3

No:-go back to main menu

4. Open file ‘record.dat’ to write

5. Assign the pointer to the end of the file to write

6. get data from user

7. write input data on a file

8. close file

9.print option to add another books

Yes:-goto step 1

No:-go back to main menu

**Edit account function**

**Steps**

1. Get account number to be edited from user
2. Open file ‘record.dat’
3. Assign the pointer to the beginning of the file to be read
4. Loop until ‘End of file ‘ is not encountered read data from file
5. a.get new data from user of that account which to be edited

b.Assign the pointer to the current position

c.Overwrite the new data on old data of that account and goto step 7

No:- Display sorry message and goto step 7

1. Close file
2. Go back to main menu

**Delete account function**

**Steps**

1.get account from user to delete

2. Declare file pointer ‘newrec’

3.open file ‘old’ to read

4. Assign the pointer to the beginning of the file to be read

5.loop until ‘End of file’ is not encountered read data from file.

6.Is user input account number= account number on a file

Yes:- a. open ‘new’ file and copy all data of ‘record’ file in ‘new’ file except that data which we want to delete

b. delete ‘record’ file and rename newrec file by ‘new’ file name and goto step 7

No:-print error message and close file and go back to main menu

7.close file

8. Go back to the main menu

**View account info function**

**Steps**

1. Open ‘record’ file
2. Assign the pointer to the beginning of the file to be read
3. Loop until ‘End of file’ is not encountered read data from file
4. Display list of all accounts with complete information
5. Close file
6. Go back to main menu

**Close the application function**

1. prompt user to close the application
2. closes the application by pressing any key….

**Sample Code**

#include<stdio.h>

#include<stdlib.h>

int i,j;

int main\_exit;

void menu();

void main();

struct date{

int month,day,year;

};

struct {

char name[60];

int acc\_no,age;

char address[60];

char citizenship[15];

double phone;

char acc\_type[10];

float amt;

struct date dob;

struct date deposit;

struct date withdraw;

}add,upd,check,rem;

void menu()

{

main();

}

void new\_acc()

{

int choice;

FILE \*ptr;

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

account\_no:

system("cls");

//printf("\t\t\t\xB2\xB2\xB2\ ADD RECORD \xB2\xB2\xB2\xB2");

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

scanf("%d/%d/%d",&add.deposit.month,&add.deposit.day,&add.deposit.year);

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

scanf("%d",&check.acc\_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);

printf("\nEnter the address:");

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

printf("\nEnter the citizenship number:");

scanf("%s",add.citizenship);

printf("\nEnter the phone number: ");

scanf("%lf",&add.phone);

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 %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(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);

if (main\_exit==1)

main();

else if(main\_exit==0)

close();

else

{

printf("\nInvalid!\a");

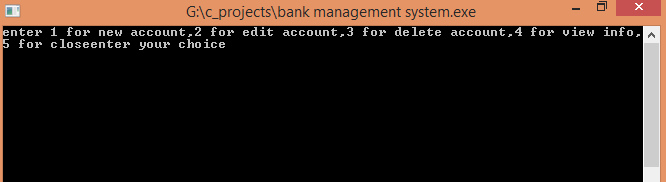
goto add\_invalid;

}

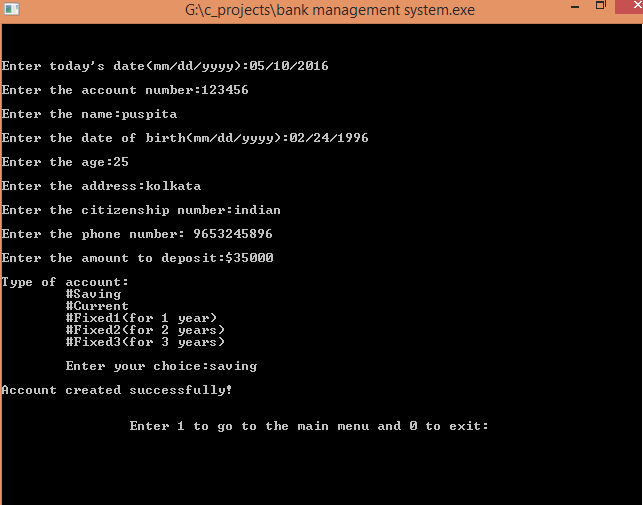
}

**SCREEN SHOTS**

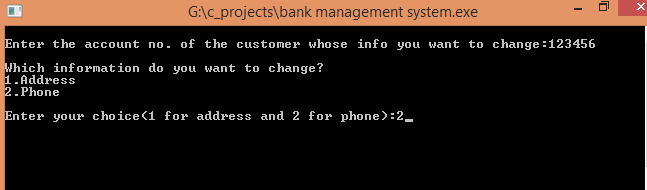
**FIRST WINDOW**

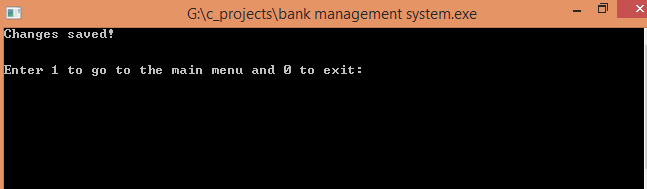
****

**ADD ACCOUNT SCREEN**

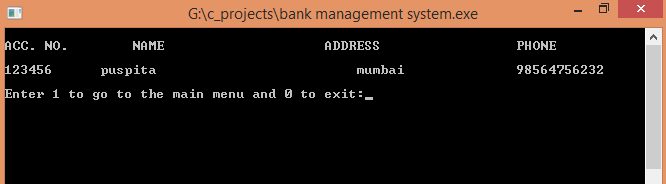
****

**EDIT ACCOUNT SCREEN**

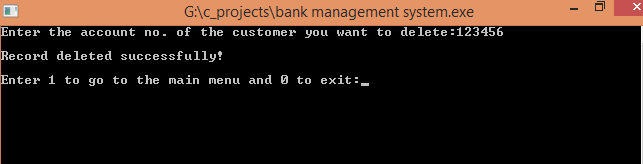
****

****

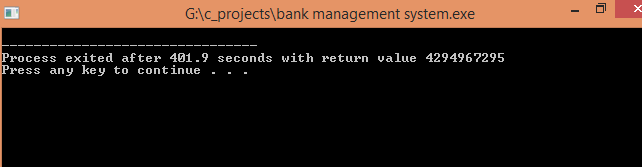
**VIEW ACCOUNT SCREEN**

****

**DELETE ACCOUNT SCREEN**

****

**CLOSE APPLICATION SCREEN**

****

# Discussion and Program Limitations

After a continuous and devoted attempt, we have finally completed our program ‘Bank Management’. The program basically gives the idea about bank management, how to manage accounts, how to edit or delete account etc. The main propose of the program is to support bank to manage it, and its accounts easily and permanently.

In my project, in future transaction history, deposit money, withdraw money etc option can be added. All concepts can be applied to make this bank management intelligent.

# Summary

Hence after the completion of the project I got familiar with the C programming and its features.

A complete and useful bank management can only be developed with lot of intensive effort and time. Due to lack of time and I am beginners in programming program that we expected can’t be developed by me. My bank management may not be must useful for bank but it will be the most useful for study and programming practice using C.

As a whole, the project has been a good learning experience for me. I have gained knowledge about the various aspects of C programming. At the same time, I have developed a deep understanding about the file handling in C.

We still want to emphasize that the program is not complete by itself. There is still a lot of room for improvement. Graphics may be added to program to make it more attractive. The mouse cursor may be initialized in order to make the program even more interactive.

# 

# References

*Balagurusamy .E., Programming In ANSI C, the Tata McGraw-Hill Companies,*

*Kanetkar Yashavant, Let Us C, BPB Publication,*

[*www.google.com*](http://www.google.com)

[*www.sourcecodesworld.com*](http://www.sourcecodesworld.com)

[*www.cprogramming.com*](http://www.cprogramming.com)