ACKNOWLEDGEMENT

I express my sincere gratitude and thanks to **Rapidcode Technologies Pvt. Ltd**. for providing me the excellent opportunity to do a project on **Bank Management System** and providing me with all the essential elements required for the completion and enhancement of this project.

I would like to thank those respondents who have taken pain in successful completion of my project work.

UNDERTAKING

My work titled **Bank Management System** as part of the Summer Internship (June – July, 2019) under the guidance of Mr. Ritesham Shastri.

If my work has been inspired by anyone else’s work, then all such work(s) has been appropriately referred by me and due acknowledgements have been made.

Any academic misconduct and dishonesty found in regard to above or otherwise shall be solely and entirely my responsibility and my faculty advisor shall not be responsible. In such a situation, I understand that a strict disciplinary action can be undertaken against me by the concerned authorities.

Name: Nishant Sharma,

Date:

Signature:

## **TABLE OF CONTENTS**

* Introduction
* Objective
* System requirement specification
* Feasibility Analysis
* Tools and Utilities
* Project Development
* Design
* Coding

**INTRODUCTION: -**

During the past several decades’ personnel function has been transformed from a relatively obscure record keeping staff to central and top level management function. There are many factors that have influenced this transformation like technological advances, professionalism, and general recognition of human beings as most important resources.

A computer based management system is designed to handle all the primary information required to calculate monthly statements of customer account which include monthly statement of any month. Separate database is maintained to handle all the details required for the correct statement calculation and generation.

This project intends to introduce more user friendliness in the various activities such as record updating, maintenance, and searching. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification or account number of that customer. Similarly, record maintenance and updating can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The entire information has maintained in the database or Files and whoever wants to retrieve can’t retrieve, only authorization user can retrieve the necessary information which can be easily be accessible from the file.

OBJECTIVE: -

A computer based management system is designed to handle all the primary information required to calculate monthly statements of customer account which include monthly statement of any month. Separate database is maintained to handle all the details required for the correct statement calculation and generation.

This project intends to introduce more user friendliness in the various activities such as record maintenance, and searching. The searching of record has been made quite simple as all the details of the customer can be obtained by simply keying in the identification or account number of that customer. Similarly, record maintenance and updating can also be accomplished by using the account number with all the details being automatically generated. These details are also being promptly automatically updated in the master file thus keeping the record absolutely up-to-date.

The main objective of our project is providing the different typed of customers’ facility, the main objective of this system is to find out the actual customer service. Etc.

This system is very easy to use, so that any user can use without getting pre-knowledge about this. It’s very much user friendly and meet almost all daily working process requirements. This system is completely CUI based and can’t be use by mouse but cab be used by keyboard.

FEASIBILITY ANALYSIS: -

Depending on the results of the initial investigation, the survey is expanded to a more detailed feasibility study. A feasibility study is a test of a system proposal. According to its workability, impact on the organization, ability to meet user’s needs and effective use of the resources its main task done during the feasibility study are: -

TECHNICAL FEASIBILITY: -

The proposed system is technically feasible as it can be developed easily with the help of available technology. The proposed system requires LINUX – red hat enterprises using VM –ware as Interface for Programming &FILE handling for storing/maintaining database.

OPERATIONAL FEASIBILITY: -

Automation makes our life easy. The proposed system is highly user friendly and is much easily able to interact with the system. Therefore, the users will readily accept the system as data entry and making queries can be easily done.

**TOOLS AND UTILITIES: -**

***GIT* - Git** is a [distributed version-control](https://en.wikipedia.org/wiki/Distributed_version_control) system for tracking changes in [source code](https://en.wikipedia.org/wiki/Source_code) during [software development](https://en.wikipedia.org/wiki/Software_development). It is designed for coordinating work among [programmers](https://en.wikipedia.org/wiki/Programmer), but it can be used to track changes in any set of [files](https://en.wikipedia.org/wiki/Computer_file). Its goals include speed, [data integrity](https://en.wikipedia.org/wiki/Data_integrity), and support for distributed, non-linear workflows.

Git was created by [Linus Torvalds](https://en.wikipedia.org/wiki/Linus_Torvalds) in 2005 for development of the [Linux kernel](https://en.wikipedia.org/wiki/Linux_kernel), with other kernel developers contributing to its initial development.[[12]](https://en.wikipedia.org/wiki/Git#cite_note-pro-git-1.2-12) Its current maintainer since 2005 is [Junio Hamano](https://en.wikipedia.org/wiki/Junio_Hamano). As with most other distributed version-control systems, and unlike most [client–server](https://en.wikipedia.org/wiki/Client%E2%80%93server) systems, every Git [directory](https://en.wikipedia.org/wiki/Directory_(computing)) on every [computer](https://en.wikipedia.org/wiki/Node_(networking)) is a full-fledged [repository](https://en.wikipedia.org/wiki/Repository_(version_control)) with complete history and full version-tracking abilities, independent of network access or a central server.[[13]](https://en.wikipedia.org/wiki/Git#cite_note-13) Git is [free and open-source software](https://en.wikipedia.org/wiki/Free_and_open-source_software) distributed under the terms of the [GNU General Public License](https://en.wikipedia.org/wiki/GNU_General_Public_License) version 2.

***GDB* -** A debugger is a program that runs other programs, allowing the user to exercise control over these programs, and to examine variables when problems arise.

GNU Debugger, which is also called **gdb,** is the most popular debugger for UNIX systems to debug C and C++ programs.

GNU Debugger helps you in getting information about the following:

* If a core dump happened, then what statement or expression did the program crash on?
* If an error occurs while executing a function, what line of the program contains the call to that function, and what are the parameters?
* What are the values of program variables at a particular point during execution of the program?
* What is the result of a particular expression in a program?

**PROJECT DEVELOPMENT: -**

**SDLC -**Software Development Life Cycle (SDLC) is a process used by the software industry to design, develop and test high quality software. The SDLC aims to produce a high-quality software that meets or exceeds customer expectations, reaches completion within times and cost estimates.

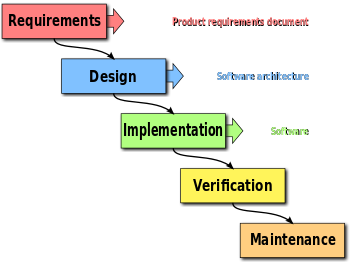
* SDLC is the acronym of Software Development Life Cycle.
* It is also called as Software Development Process.
* SDLC is a framework defining tasks performed at each step in the software development process.
* ISO/IEC 12207 is an international standard for software life-cycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

****

**WATERFALL MODEL: -**

The **waterfall model** is a breakdown of project activities into linear [sequential](https://en.wikipedia.org/wiki/Sequence) phases, where each phase depends on the deliverables of the previous one and corresponds to a specialisation of tasks. The approach is typical for certain areas of [engineering design](https://en.wikipedia.org/wiki/Engineering_design). In [software development](https://en.wikipedia.org/wiki/Software_development_process), it tends to be among the less iterative and flexible approaches, as progress flows in largely one direction ("downwards" like a [waterfall](https://en.wikipedia.org/wiki/Waterfall)) through the initiation, [analysis](https://en.wikipedia.org/wiki/Analysis), [design](https://en.wikipedia.org/wiki/Software_design), [construction](https://en.wikipedia.org/wiki/Software_construction), [testing](https://en.wikipedia.org/wiki/Software_testing), [deployment](https://en.wikipedia.org/wiki/Implementation) and [maintenance](https://en.wikipedia.org/wiki/Software_maintenance).

The waterfall development model originated in the [manufacturing](https://en.wikipedia.org/wiki/Manufacturing) and [construction](https://en.wikipedia.org/wiki/Construction) industries; where the highly structured physical environments meant that design changes became prohibitively expensive much sooner in the development process. When first adopted for software development, there were no recognised alternatives for knowledge-based creative work.

****

SYSYTEM REQUIREMENTS:

Hardware specifications: -

Hardware is a set of physical components, which performs the functions of applying appropriate, predefined instructions. In other words, one can say that electronic and mechanical parts of computer constitute hardware.

This package is designed on a powerful programming language C. It is a powerful CUI. . It can run on almost all the popular microcomputers. The following are the minimum hardware specifications to run this package: -

Personal Computer: -

It minimum contains P-IV processor

Processor with 512 MB RAM

Software Requirements**: -**

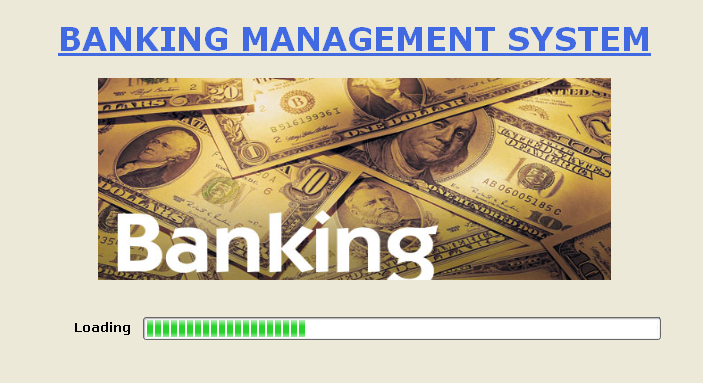
The software is a set of procedures of coded information or a program which when fed into the computer hardware, enables the computer to perform the various tasks. Software is like a current inside the wire, which cannot be seen but its effect can be felt.

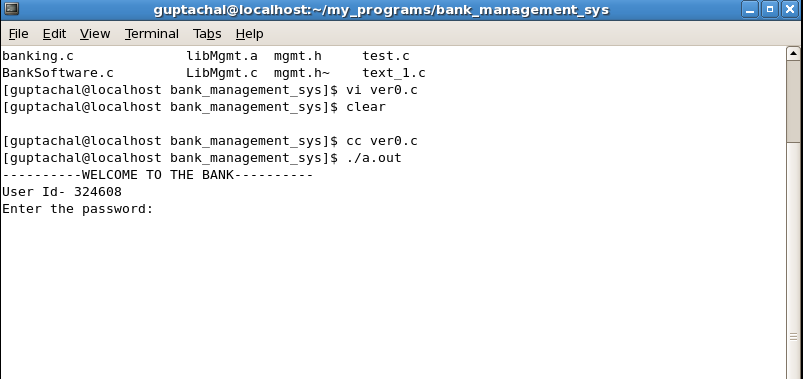
1. Operating System-UNIX/LINUX

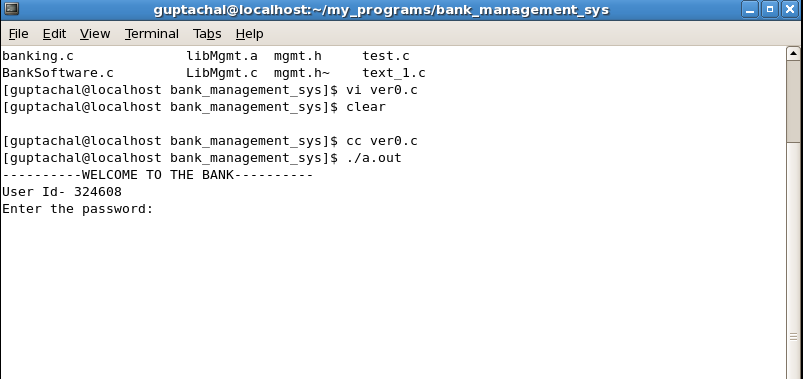
2. Application Software: - VM Ware

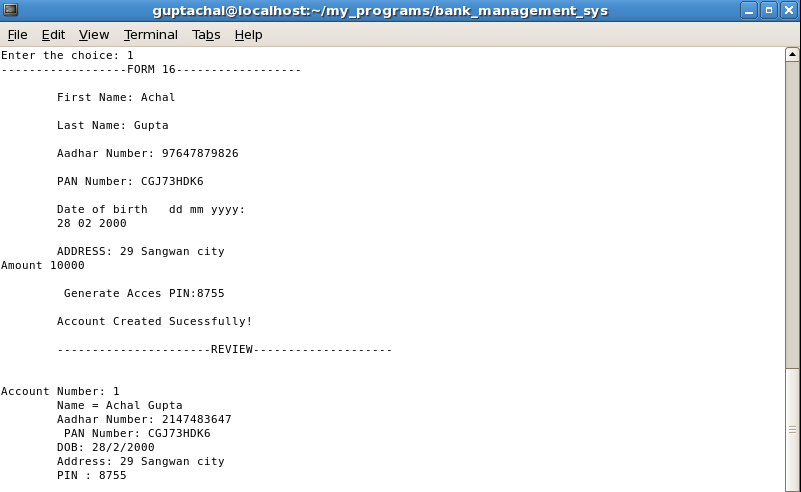
3. Editor: - vi editor

**DESIGN AND SNAPSHOT: -**

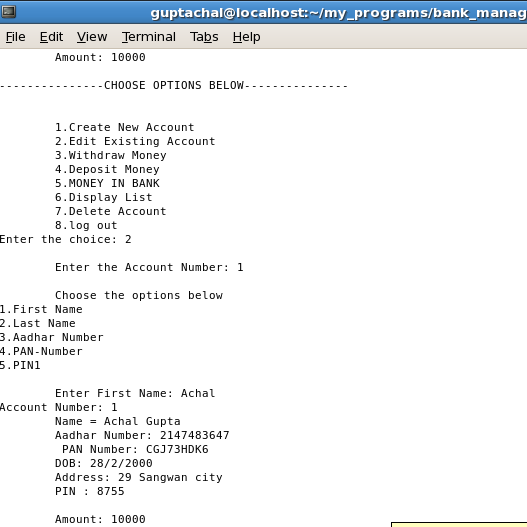
* SPLASH FORM
* 
* LOGIN FORM



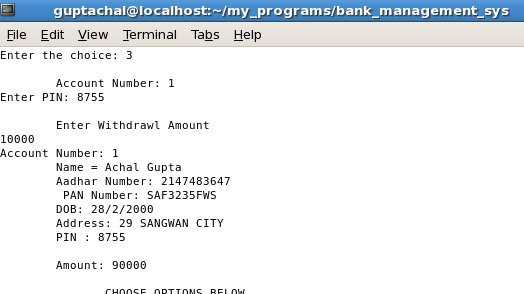
* MID FORM
* 
* ACCOUNT OPENING FORM



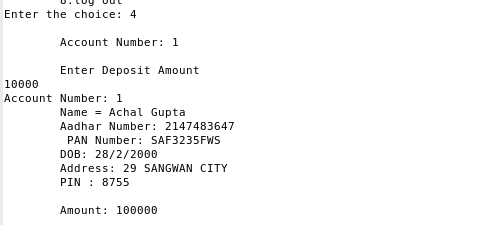
* ACCOUNT EDIT FORM



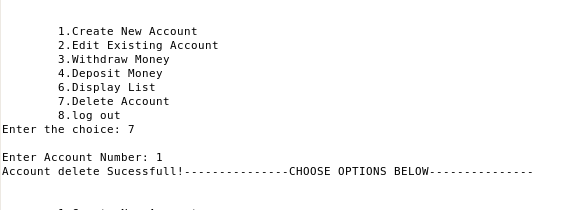
WITHDRAL FORM



* DEPOSIT FORM



ACCOUNT CLOSING FORM



**CODING: -**

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#define SIZE 50

int pass\_verification(char\*,char\*);

char\* pass\_saved="wo ai ni";

struct create\_account

{

char FirstName[15];

char LastName[15];

long int AadharNo;

char PanNumber[15];

int dd,mm,yyyy;

char House[10];

char Locality[20];

char City[20];

int Amount;

char pin[6];

};

int i=0;

typedef struct create\_account form;

void Delete\_Account(void);

void Generate\_Account(void);

void Update\_Account(void);

void Display(int);

form detail[100];

int Acc\_No=0;

void Display\_List(void);

void delay(void);

void update(int,int);

void print(void);

int main()

{

int confirm,choice,j;

char \*pass=(char\*)malloc(SIZE);

printf("----------WELCOME TO THE BANK----------\n");

printf("User Id- 324608\n");

there:

printf("Enter the password: ");

fgets(pass,SIZE,stdin);

\*(pass+strlen(pass)-1)='\0';

confirm=pass\_verification(pass,pass\_saved);

if(confirm==1)

{

printf("Password Verified\n");

while(1)

{

printf("---------------CHOOSE OPTIONS BELOW---------------\n");

printf("\n\n");

x:

printf("\t1.Create New Account\n");

printf("\t2.Edit Existing Account\n");

printf("\t3.Withdraw Money\n");

printf("\t4.Deposit Money\n");

printf("\t5.MONEY IN BANK\n");

printf("\t6.Display List\n");

printf("\t7.Delete Account\n");

printf("\t8.log out\n");

printf("Enter the choice: ");

scanf("%d",&choice);

switch(choice)

{

case 1:

Acc\_No++;

i++;

Generate\_Account();

break;

case 2:Update\_Account();

break;

/\* case 3:Withdrawl();

break;

case 4:Deposit();

break;

case 5:MoneyInBank();

break;\*/

case 6:

for(j=0;j<Acc\_No;j++)

Display(j);

delay();

break;

case 7:Delete\_Account();

break;

default:printf("\n--------Invalid Choice-------\n");

goto x;

}

}

}

else

{

printf("WRONG PASSWORD\n");

goto there;

}

return 0;

}

int pass\_verification(char\* password,char\* pass\_pre)

{

while(\*password!='\0'||\*pass\_pre!='\0')

{

if(\*password!=\*pass\_pre)

return 0;

password++;

pass\_pre++;

}

if(\*password=='\0'&&\*pass\_pre=='\0')

return 1;

}

void Generate\_Account()

{

FILE \*fptr;

fptr=(fopen("data.txt","w+"));

if(fptr==NULL)

{

printf("Error!");

}

printf("------------------FORM 16------------------");

printf("\n\n\tFirst Name: ");

scanf("%s",detail[i].FirstName);

printf("\n\tLast Name: ");

scanf("%s",detail[i].LastName);

printf("\n\tAadhar Number: ");

scanf("%ld",&detail[i].AadharNo);

{

printf("\n\tInvalidAadhar Number \n");

}

printf("\n\tPAN Number: ");

scanf("%s",detail[i].PanNumber);

printf("\n\tDate of birth dd mm yyyy: ");

printf("\n\t");

scanf("%d",&detail[i].dd);

scanf("%d",&detail[i].mm);

scanf("%d",&detail[i].yyyy);

fflush(stdin);

printf("\n\tADDRESS: ");

// printf("House Number: ");

scanf("%s",detail[i].House);

fflush(stdin);

scanf("%s",detail[i].Locality);

fflush(stdin);

scanf("%s",detail[i].City);

fflush(stdin);

printf("Amount ");

scanf("%d",&detail[i].Amount);

fflush(stdin);

printf("\n\t Generate Acces PIN:");

scanf("%s",&detail[i].pin);

print();

}

//printing all the details in a file

void print()

{

FILE \*fptr;

fptr=(fopen("data.txt","w+"));

if(fptr==NULL)

{

printf("Error!");

}

for(i=0;i<Acc\_No;i++);

fprintf(fptr,"%d",Acc\_No);

fprintf(fptr,"%s%s",detail[i].FirstName,detail[i].LastName);

fprintf(fptr,"%d%s",detail[i].AadharNo,detail[i].PanNumber);

fprintf(fptr,"%d/%d/%d",detail[i].dd,detail[i].mm,detail[i].yyyy);

fprintf(fptr,"%s%s%s%s",detail[i].House,detail[i].Locality,detail[i].City,detail[i].pin);

// fprintf(fptr,"%d",detail[i].pin);

fprintf(fptr,"%d",detail[i].Amount);

fclose(fptr);

printf("\n\tAccount Created Sucessfully!\n");

printf("\n\t----------------------REVIEW--------------------\n\n\n");

Display(i);

delay();

}

void Display(int j)

{

printf("Account Number: %d \n",j);

if(pass\_verification(detail[j].FirstName,"null"))

{

printf("Account doesn't Exist!\n");

}

else

{

printf("\tName = %s %s",detail[j].FirstName,detail[j].LastName);

printf("\n\tAadhar Number: %d \n\t PAN Number: %s",detail[j].AadharNo,detail[j].PanNumber);

printf("\n\tDOB: %d/%d/%d",detail[j].dd,detail[j].mm,detail[j].yyyy);

printf("\n\tAddress: %s %s %s\n ",detail[j].House,detail[j].Locality,detail[j].City);

printf("\tPIN : %s\n",detail[j].pin);

printf("\n\tAmount: %d\n\n",detail[j].Amount);

}

delay();

}

void Delete\_Account(void)

{

int del,new;

printf("\nEnter Account Number: ");

scanf("%d",&del);

new=del-1;

strcpy(detail[new].FirstName,"null");

printf("Account delete Sucessfull!");

delay();

}

void delay(void)

{

long int i;

for(i=0;i<1000000000;i++);

}

void Update\_Account(void)

{

int acc,choice;

printf("\n\tEnter the Account Number: ");

scanf("%d",&acc);

printf("\n\tChoose the options below");

printf("\n1.First Name\n2.Last Name\n3.Aadhar Number\n4.PAN-Number\n5.PIN");

scanf("%d",&choice);

update(acc,choice);

}

void Withdrawl()

{

char \*p=(char\*)malloc(6);

int ac;

printf("\n\tAccount Number: ");

scanf("%d",&ac);

x:

printf("Enter PIN: ");

scanf("%s",p);

if(pass\_verification(detail[ac].pin,p)==1)

{

if(detail[i].Amount<=1000)

{

printf("Insufficient Balancse!");

return;

}

else

{

update(ac,-1);

}

}

else {

printf("Invalid PIN!\n");

goto x;

}

}

void Deposit(void)

{

int ac;

printf("\n\tAccount Number: ");

scanf("%d",&ac);

update(ac,0);

}

void update(int acc , int choice)

{

char \*up=(char\*)malloc(20);

FILE \*old,\*new;

old=fopen("data.txt","r+");

new=fopen("data.txt","w");

int amt;

switch(choice)

{

case -1:

printf("\n\tEnterWithdrawl Amount\n");

scanf("%d",&amt);

detail[acc].Amount=detail[acc].Amount-amt;

break;

case 0:

printf("\n\tEnter Deposit Amount\n");

scanf("%d",&amt);

detail[acc].Amount+=amt;

break;

case 1:

printf("\n\tEnter First Name: ");

scanf("%s",up);

strcpy(detail[acc].FirstName,up);

break;

case 2:

printf("\n\tLast Name: ");

scanf("%s",up);

strcpy(detail[acc].LastName,up);

break;

case 3:

printf("\n\tAadhar Number: ");

scanf("%d",&detail[acc].AadharNo);

break;

case 4:

printf("\n\tPAN Number");

scanf("%s",detail[acc].PanNumber);

break;

default: printf("\n\tInvalid Choice!");

}

Display(acc);

}

BIBLIOGRAPHY AND REFERENCES:

BOOK: - EXPLORING –C, THE C PROGRAMMING LANGUAGE

WEBSITES: - [WWW.RAPIDCODE.COM](http://WWW.RAPIDCODE.COM)

SEARCH ENGINES: - YAHOO, MSN, GOOGLE etc.