**Introduction:**

* Bank Management system is used to save the records of students, clients, employee in Bank. Bank management system is an application for maintaining a personal account in a bank, and the system provides the access to the customer to create an account, deposit, withdraw the cash from his or her account, also to view reports of all accounts present he or her had.

**Research:**

* This is a software for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks.

**Features:**

* In my Project there are some features which will be easy to the customer to store the data in their application.
* Adding multiple customer data in the record.
* Showing the List of all the accounts of customer and their data of acc\_no and balance in the record.
* Deposit & Withdrawing of money
* Exit the application.

**Requirements:**

* visual code studio with C compiler
* Github (push the code)

**Swot analysis:**

* ->strength \*Banking Industry is the Oldest Industry so this application will help to make changes drastically. \*Add unlimited number of customer details in this application
* After adding a particular customer details we can continue to add another customer details with yes/No option.
* Time saving for an organization to check for the customer data.
* this application will update the data and show the output in a phased manner
* Exit the application in an easier manner by providing Yes/No option

**Weakness:**

* it takes time to update the modified details.

**4W's and 1H:**

* Who: The IT team of an organization will work on this Banking Management system
* What: To store customer data in an application
* Where: In an application software
* How: By selecting the neccesary option we can enter the input of an customer and store the information in an application

**Detail requirements:**

* High level requirements:-
* Select anyone of the following:
* Add details
* Store details
* Exit
* low level requirements:-
* Enter the following information to the customer
* Enter the name of customer
* Enter the balance of the employee

### Source Code:-

#include <stdio.h>

struct custmr

{

int acc\_no;

char name[80];

int balance;

};

void accept(struct custmr[], int);

void display(struct custmr[], int);

int search(struct custmr[], int, int);

void deposit(struct custmr[], int, int, int);

void withdraw(struct custmr[], int, int, int);

int main()

{

struct custmr data[20];

int n, choice, acc\_no, amount, index;

printf("Banking System\n\n");

printf("Num of custmr records you want to enter? : ");

scanf("%d", &n);

accept(data, n);

do

{

printf("\nBanking System Menu :\n");

printf("Press 1 to display all records.\n");

printf("Press 2 to search a record.\n");

printf("Press 3 to deposit amount.\n");

printf("Press 4 to withdraw amount.\n");

printf("Press 0 to exit\n");

printf("\nEnter choice(0-4) : ");

scanf("%d", &choice);

switch (choice)

{

case 1:

display(data, n);

break;

case 2:

printf("Enter acc num to search : ");

scanf("%d", &acc\_no);

index = search(data, n, acc\_no);

if (index == - 1)

{

printf("Record not found : ");

}

else

{

printf("A/c Num: %d\nName: %s\nBalance: %d\n",

data[index].acc\_no, data[index].name,

data[index].balance);

}

break;

case 3:

printf("Enter acc num : ");

scanf("%d", &acc\_no);

printf("Enter amount to deposit : ");

scanf("%d", &amount);

deposit(data, n, acc\_no, amount);

break;

case 4:

printf("Enter acc num : ");

scanf("%d", &acc\_no);

printf("Enter amount to withdraw : ");

scanf("%d", &amount);

withdraw(data, n, acc\_no, amount);

}

}

while (choice != 0);

return 0;

}

void accept(struct custmr list[80], int s)

{

int i;

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

{

printf("\nEnter data for Record #%d", i + 1);

printf("\nEnter acc\_no : ");

scanf("%d", &list[i].acc\_no);

fflush(stdin);

printf("Enter name : ");

gets(list[i].name);

list[i].balance = 0;

}

}

void display(struct custmr list[80], int s)

{

int i;

printf("\n\nA/c No\tName\tBalance\n");

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

{

printf("%d\t%s\t%d\n", list[i].acc\_no, list[i].name,

list[i].balance);

}

}

int search(struct custmr list[80], int s, int num)

{

int i;

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

{

if (list[i].acc\_no == num)

{

return i;

}

}

return - 1;

}

void deposit(struct custmr list[], int s, int num, int amt)

{

int i = search(list, s, num);

if (i == - 1)

{

printf("Record not found");

}

else

{

list[i].balance += amt;

}

}

void withdraw(struct custmr list[], int s, int num, int amt)

{

int i = search(list, s, num);

if (i == - 1)

{

printf("Record not found\n");

}

else if (list[i].balance < amt)

{

printf("Insufficient balance\n");

}

else

{

list[i].balance -= amt;

}

}

Output:

