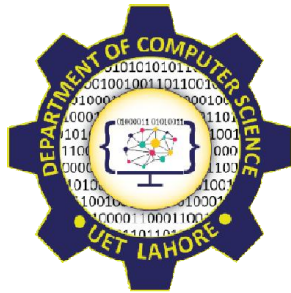


HEAVEN'S BANK



Session 2023-2027

Submitted by:

Amir Hashmi

2023-CS-11

Supervised by:

Dr. Muhammad Owais Hassan

Course:

CSC-102 Programming Fundamentals

Department of Computer Science

University of Engineering and Technology

Lahore Pakistan

- **Short Description:**

- I am trying to solve a problem related to online banking in this application and the main objective is to provide an online platform for banking transactions. In the field of computer science it provides an easy way to manage banking transactions.
- The console interface is designed to be easily understandable for users, they do not require any additional information before using this software.
- At the end of this project I expect to deliver user an application through which user can easily manage his/her banking transactions i.e open account, deposit, withdraw and transfer money. The user can get debit card and can get or pay loans.

- **Users of Application:**

There are two types of users in this application. An Admin who have the authority over the application and the Client who can access his/her account info and conduct different transactions easily.

- **Manager**

The admin or manager holds full authority over the application. They can add or remove clients, view client records, approve loans, inspect the assets held by the bank, and even add new assets if necessary.

- **Client**

The client has the capability to register their account, enabling them to perform various transactions such as depositing, transferring, withdrawing, or investing money in gold. Additionally, they have the ability to view a record of their transactions and possess the authority to modify their personal information.

• Functional Requirements

As a	I want to perform	So that I can
Manager	Open Account	Open the account of client
	Remove Account	Remove the account of client
	View Accounts	View the accounts of Clients
	View Total balance	Total balance of bank
	View complaints	View the complaints of clients
	Block Debit Card	Block debit card of client
	Hire Staff	Hire the staff for bank
	Expel Staff	Expel staff of bank
	View Staff	View staff of bank
	Log out	Log out from the Application
Client	View Account Status	View the status of my account
	Deposit Money	Deposit money in my account
	Withdraw Money	Withdraw money from my account
	Transfer Donations	Transfer Donations to any welfare organization
	Get Debit Card	Get debit card
	Get Loan	Get loan from bank
	Exchange Currency	Exchange my currency
	Pay Loan	Pay my loan.
	Submit Complaints	Submit my complaints
	Log Out	Log out from the application

- **Wireframes**



Figure1:Login Screen

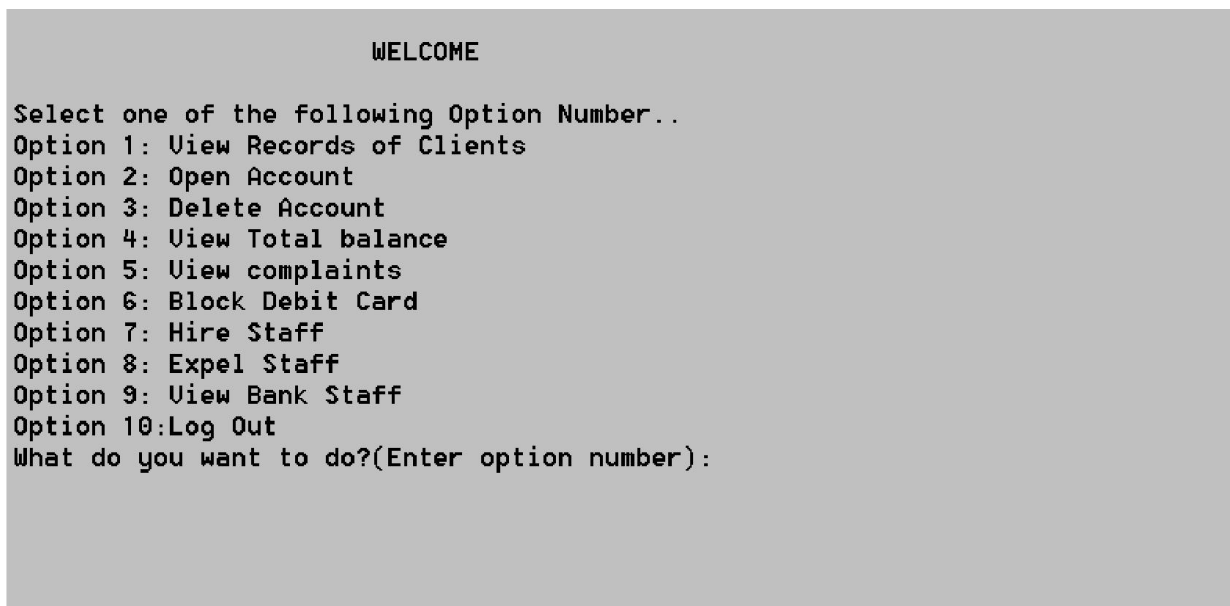


Figure2: Manager Main Menu Screen

```
WELCOME

Select one of the following Option Number..
Option 1: View Account Status
Option 2: Deposit Money
Option 3: Withdraw Money
Option 4: Transfer Donations
Option 5: Get Debit Card
Option 6: Get Loan
Option 7: Pay Loan
Option 8: Exchange Currency
Option 9: Submit Complaints
Option 10: Log Out
What do you want to do?(Enter option number):
```

Figure3: Client Main Menu Screen

- **Data Structures (Parallel Arrays)**

- The Parallel arrays used in my project are:

1. string name[10];
2. string staff[10];
3. string des[10];
4. int password[10];
5. string role[10];
6. int balance[10]={0,0,0,0,0,0,0,0,0,0};
7. int loan[10]={0,0,0,0,0,0,0,0,0,0};
8. debit[10]={"N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A"};
9. string complaints[10];

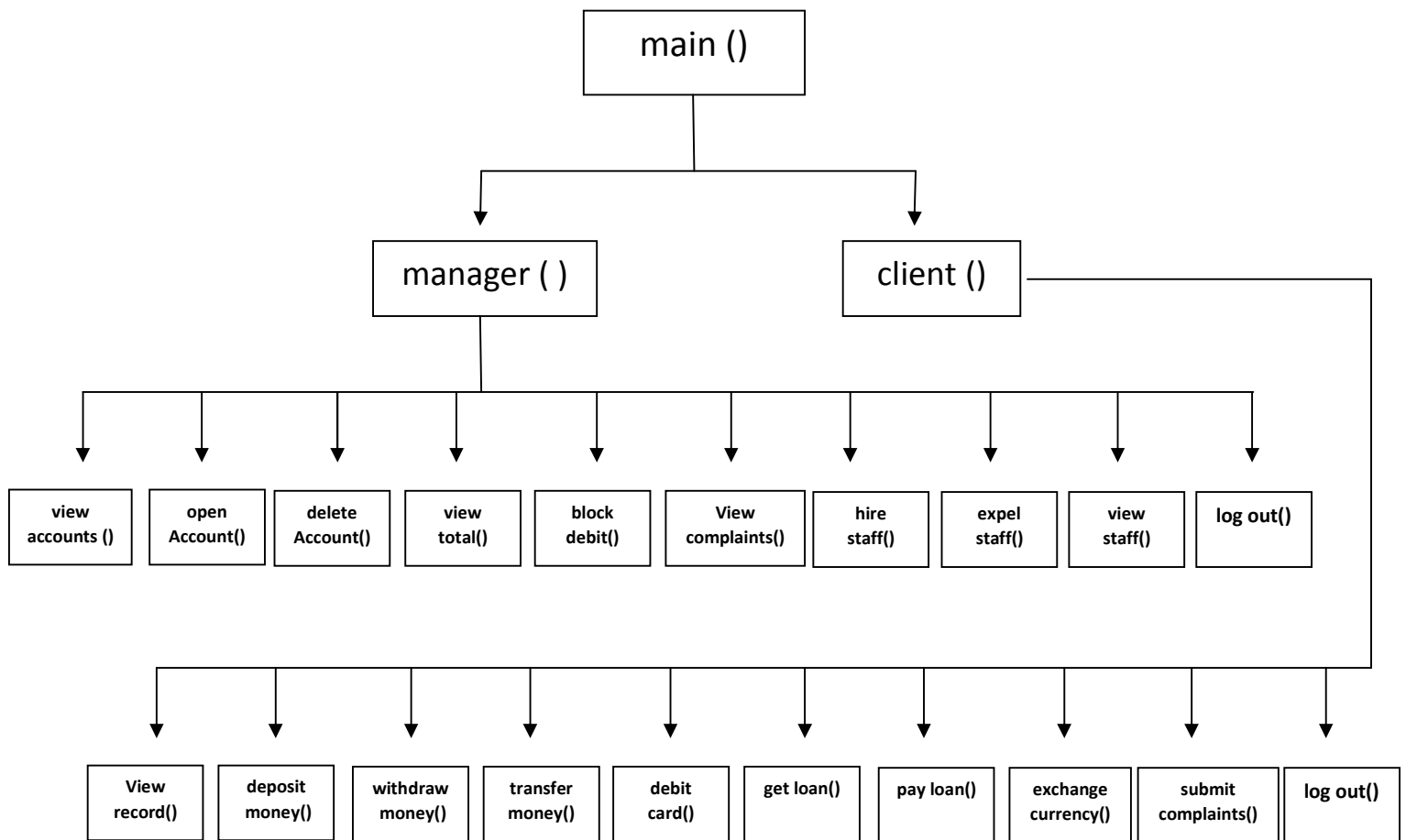
- The counter variables used for parallel arrays are:

1. int s=0;
2. int i=0;
3. int c=0;

• Function Prototypes

1. `bool signinhead (string name[],int password[],string role[],int &option,int &s,int &i,int &c);`
2. `bool signin (string name[],int password[],string role[],int &s,int &i,int &c,string signinname,int signinpassword);`
3. `void signuphead(string name[],int password[],string role[],int &option,int &s,int &i,int &c);`
4. `void signup (string name[],int password[],string role[],int &s,int &i,int &c,string signupname,int signuppassword);`
5. `int manager ();`
6. `int client ();`
7. `bool viewaccounts (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
8. `void viewtotal (string name[],int password[],string role[],int balance[],int loan[],int &s,int &i,int &c);`
9. `bool openaccount (string name[],int password[],string role[],int &s,int &i,int &c);`
10. `bool deleteaccount (string name[],int password[],string role[],int balance[],int &s,int &i,int &c);`
11. `bool hirestaff (string staff[],string des[],int &s,int &i,int &c);`
12. `void header ();`
13. `bool expelstaff (string staff[],string des[],int &s,int &i,int &c);`
14. `bool viewstaff (string staff[],string des[],int &s,int &i,int &c);`
15. `bool blockdebit (string name[],int password[],string role[],string debit[],int &s,int &i,int &c);`
16. `bool viewcomplaints (string name[],int password[],string role[],string complaints[],int &s,int &i,int &c);`
17. `void viewrecord (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
18. `bool depositmoney (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
19. `bool withdrawmoney (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
20. `void exchangecurrency ();`
21. `bool transfermoney (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],string &welfare,int &s,int &i,int &c);`
22. `bool debitcard (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
23. `bool getloan (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
24. `void payloan (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
25. `bool submitcomplaint (string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);`
26. `bool exit ();`

- **Functions Working Flow**



Complete Code of the Business Application

```
#include<iostream>

#include<conio.h>

#include<windows.h>

#include<iomanip>

using namespace std;

bool signinhead(string name[],int password[],string role[],int &option,int &s,int &i,int &c);

bool signin(string name[],int password[],string role[],int &s,int &i,int &c,string signinname,int signinpassword);

void signuphead(string name[],int password[],string role[],int &option,int &s,int &i,int &c);

void signup(string name[],int password[],string role[],int &s,int &i,int &c,string signupname,int signuppassword);

int manager();

int client();

bool viewaccounts(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);

void viewtotal(string name[],int password[],string role[],int balance[],int loan[],int &s,int &i,int &c);

bool openaccount(string name[],int password[],string role[],int &s,int &i,int &c);

bool deleteaccount(string name[],int password[],string role[],int balance[],int &s,int &i,int &c);

bool hirestaff(string staff[],string des[],int &s,int &i,int &c);

void header();

bool expelstaff(string staff[],string des[],int &s,int &i,int &c);

bool viewstaff(string staff[],string des[],int &s,int &i,int &c);

bool blockdebit(string name[],int password[],string role[],string debit[],int &s,int &i,int &c);

bool viewcomplaints(string name[],int password[],string role[],string complaints[],int &s,int &i,int &c);

void viewrecord(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);

bool depositmoney(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);

bool withdrawmoney(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c);
```



```
void exchangecurrency();

bool transfermoney(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string
debit[],string &welfare,int &s,int &i,int &c);

bool debitcard(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int
&s,int &i,int &c);

bool getloan(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int
&s,int &i,int &c);

void payloan(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int
&s,int &i,int &c);

bool submitcomplaint(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string
debit[],int &s,int &i,int &c);

bool exit();

main()
{
int s=0;
string name[10];
int i=0;
int c=0;
int resc;
int resm;
string welfare;
int option;
string staff[10];
string des[10];
int password[10];
string role[10];
int balance[10]={0,0,0,0,0,0,0,0,0,0};
int loan[10]={0,0,0,0,0,0,0,0,0,0};
string debit[10]={"N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A","N/A"};
string complaints[10];
```

//This loop will continuously display user their respective menu and take input from user until the user press exit.

```
while(1)
{
system("cls");
header();
cout<<"\n 1. SIGN IN\n";
cout<<" 2. SIGN UP \n";
cout<<" 3. EXIT \n";
cout<<" Enter Option: ";
cin>>option;
system("cls");
if(option==1)
{
//This if statement will take user to sign in menu.
if(signinhead(name,password,role,option,s,i,c))
{
cout<<" Congratulations! "<<name[c] <<" You are signed in successfully.";
if(role[c]=="Client" || role[c]=="client")
{
//This if statement will take user to Client menu.
Sleep(1500);
system("cls");
while(true)
{
resc=client();
if(resc==1)
{
system("cls");
cout<<"\t\t\t\t View Account Status \n\n";
```

```
viewrecord(name,password,role,balance,loan,complaints,debit,s,i,c);
}
if(resc==2)
{
system("cls");
cout<<"\t\t\t\t Deposit Money \n\n";
if(depositmoney(name,password,role,balance,loan,complaints,debit,s,i,c))
cout<<" Your Amount is successfully deposited.Now your Current Balance is: "<<balance[c];
}
if(resc==3)
{
system("cls");
cout<<"\t\t\t\t Withdraw Money \n\n";
if(withdrawmoney(name,password,role,balance,loan,complaints,debit,s,i,c))
cout<<" Your Amount is successfully withdraw.Now your Current Balance is: "<<balance[c];
else
cout<<" Sorry! You don't have enough balance.";
}
if(resc==4)
{
system("cls");
cout<<"\t\t\t\t Transfer Donations \n\n";
if(transfermoney(name,password,role,balance,loan,complaints,debit,welfare,s,i,c))
cout<<" Your donations is successfully transfered to "<<welfare<<". Now your Current Balance is: "<<balance[c];
else
cout<<" Sorry! You don't have enough balance.";
}
if(resc==5)
```

```
{
system("cls");
cout<<"\t\t\t\t Get Debit Card \n\n";
if(debitcard(name,password,role,balance,loan,complaints,debit,s,i,c))
cout<<" Congratulations!Your "<<debit[c]<<" card has been issued.";
else
cout<<" Sorry! Your monthly salary is insufficient.";
}
if(resc==6)
{
system("cls");
cout<<"\t\t\t\t Get Loan \n\n";
if(getloan(name,password,role,balance,loan,complaints,debit,s,i,c))
cout<<" Congratulations!Your "<<loan[c]<<" loan has been approved.";
else
cout<<" Sorry! Your monthly salary is insufficient.";
}
if(resc==7)
{
system("cls");
cout<<"\t\t\t\t Pay Loan \n\n";
payloan(name,password,role,balance,loan,complaints,debit,s,i,c);
}
if(resc==8)
{
system("cls");
cout<<"\t\t\t\t Exchange Currency \n\n";
exchangecurrency();
}
```

```
if(resc==9)
{
system("cls");
cout<<"\t\t\t Submit Complaints \n\n";
if(submitcomplaint(name,password,role,balance,loan,complaints,debit,s,i,c))
cout<<" Your Complaint has been submitted. We will try to resolve it as early as possible.";
}
if(resc==10)
{
system("cls");
if(exit())
break;
}
cout<<" \n\n Press any key to continue: "<<endl;
getch();
system("cls");
}
}
else if(role[c]=="Manager" | |role[c]=="manager")
{
//This if statement will take user to Manager menu.
Sleep(1500);
system("cls");
while(true)
{
resm=manager();
if(resm==1)
{
```

```
system("cls");

cout<<"\t\t\t\t View Records Of Clients \n\n";

if(viewaccounts(name,password,role,balance,loan,complaints,debit,s,i,c))

cout<<" No Clients available.";

}

if(resm==2)

{

system("cls");

cout<<"\t\t\t\t Open Account \n\n";

if(openaccount(name,password,role,s,i,c))

cout<<" Account of Client "<<name[c]<<" is successfully Opened. ";

}

if(resm==3)

{

system("cls");

cout<<"\t\t\t\t Delete Account \n\n";

if(deleteaccount(name,password,role,balance,s,i,c))

cout<<" Sorry! The client you want to delete does not exist.";

}

if(resm==4)

{

system("cls");

cout<<"\t\t\t\t View Total Balance \n\n";

viewtotal(name,password,role,balance,loan,s,i,c);

}

if(resm==5)

{

system("cls");

cout<<"\t\t\t\t View Complaints \n\n";
```

```
if(viewcomplaints(name,password,role,complaints,s,i,c))
cout<<" No Complaints available.";
}
if(resm==6)
{
system("cls");
cout<<"\t\t\t\t Block Debit Card \n\n";
if(blockdebit(name,password,role,debit,s,i,c))
cout<<" The Client whose debit card you want to block does not exist.";
}
if(resm==7)
{
system("cls");
cout<<"\t\t\t\t Hire Staff \n\n";
if(hirestaff(staff,des,s,i,c))
cout<<" Employee "<<staff[s]<<" is successfully hired. ";
}
if(resm==8)
{
system("cls");
cout<<"\t\t\t\t Expel Staff \n\n";
if(expelstaff(staff,des,s,i,c))
cout<<" Sorry! The employee you want to expell does not exist.";
}
if(resm==9)
{
system("cls");
cout<<"\t\t\t\t View Bank Staff \n\n";
```

```
if(viewstaff(staff,des,s,i,c))
cout<<" No staff available.";
}
if(resm==10)
{
system("cls");
exit();
break;
}
cout<<" \n\n Press any key to continue: "<<endl;
getch();
system("cls");
}
}
}
else
{
cout<<" You don't have an account please sign up.";
Sleep(1500);
}
}
else if(option==2)
{
//This if statement will take user to sign up menu.
signuphead(name,password,role,option,s,i,c);
}
else if(option==3)
{
//This if statement will terminate the program.
```

Aamir Hashmi (2023-CS-11)CSC-102 Programming Fundamentals

```
        if(signin(name,password,role,s,i,c,signinname,signinpassword)){
            return true;}

        else{
            return false;}

    }

//This function will sign in the user.

bool signin(string name[],int password[],string role[],int &s,int &i,int &c,string signinname,int signinpassword)
{

    int x=0;
    for(int n=0;n<10;n++)
    {
        if(name[n]==signinname&&password[n]==signinpassword)
        {
            c=n;
            return true;

            x=1;
            break;
        }
    }

    if(x==0)
    {
        return false;
    }
}

//This function will take username and password from user and passes it to sign in function.

void signuphead(string name[],int password[],string role[],int &option,int &s,int &i,int &c)
{

    int count=1;
```

```
string signupname;
int signuppasword;
cout<<"\t\t\t SIGN UP \n";
cout<<" Enter UserName: ";
cin>>signupname;
cout<<" Enter Password: ";
cin>>signuppasword;
for(int l=0;l<10;l++)
{
if(name[l]==signupname)
{
cout<<" Sorry this account already exist.";
Sleep(1500);
count=0;
break;
}
}
if(count==1)
{
cout<<" Enter your role (Manager/Client): ";
cin>>role[i];
signup(name,password,role,s,i,c,signupname,signuppasword);
cout<<" Congratulations "<<name[i]<<"! You are sucessfully signed up.";
Sleep(1500);
i=i+1;
}
}
```

//This function will sign in the user.

```
void signup(string name[],int password[],string role[],int &s,int &i,int &c,string signupname,int signuppasword)
{
    name[i]=signupname;
    password[i]=signuppasword;
}
```

//This function will display tasks that manager can perform,takes input from manager and returns it to main function.

```
int manager()
{
    int choose;
    cout<<"\n\t\tWELCOME\n\n";
    cout<<" Select one of the following Option Number..\n";
    cout<<" Option 1: View Records of Clients"<<endl;
    cout<<" Option 2: Open Account"<<endl;
    cout<<" Option 3: Delete Account"<<endl;
    cout<<" Option 4: View Total balance"<<endl;
    cout<<" Option 5: View complaints"<<endl;
    cout<<" Option 6: Block Debit Card"<<endl;
    cout<<" Option 7: Hire Staff"<<endl;
    cout<<" Option 8: Expel Staff"<<endl;
    cout<<" Option 9: View Bank Staff"<<endl;
    cout<<" Option 10:Log Out"<<endl;
    cout<<" What do you want to do?(Enter option number):";
    cin>>choose;
    return choose;
}
```

//This function will display tasks that client can perform,takes input from client and returns it to main function.

```
int client()
```

```
{
    int choice;
    cout<<"\t\tWELCOME\n\n";
    cout<<" Select one of the following Option Number..\n";
    cout<<" Option 1: View Account Status"<<endl;
    cout<<" Option 2: Deposit Money"<<endl;
    cout<<" Option 3: Withdraw Money"<<endl;
    cout<<" Option 4: Transfer Donations"<<endl;
    cout<<" Option 5: Get Debit Card"<<endl;
    cout<<" Option 6: Get Loan"<<endl;
    cout<<" Option 7: Pay Loan"<<endl;
    cout<<" Option 8: Exchange Currency"<<endl;
    cout<<" Option 9: Submit Complaints"<<endl;
    cout<<" Option 10:Log Out"<<endl;
    cout<<" What do you want to do?(Enter option number):";
    cin>>choice;
    return choice;
}

//This function will show the accounts details of all clients in bank to manager.
bool viewaccounts(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string
debit[],int &s,int &i,int &c)
{
    int count=0;
    for(int x=0;name[x]!='\0';x++)
    {
        if(role[x]=="Client"&&name[x]!=" ")
            count++;
    }
    if(count!=0)
```

```
{
    cout<<" NAME\t\tCurrent Balance\t\tDebit Card\t\tLoan"<<endl;
    for(int i=0;name[i]!='\0';i++)
    {
        if(role[i]=="Client"&&name[i]!=" ")
            cout<<" "<<name[i]<<"\t"<<balance[i]<<"\t\t"<<debit[i]<<"\t\t"<<loan[i]<<endl;
    }
}

else
    return true;
}

//This function will show client his account details.

void viewrecord(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string
debit[],int &s,int &i,int &c)
{
    cout<<" NAME\t\tCurrent Balance\t\tDebit Card\t\tLoan"<<endl;
    cout<<" "<<name[c]<<"\t"<<balance[c]<<"\t\t"<<debit[c]<<"\t\t"<<loan[c];
}

//Client will use this function to deposit money.

bool depositmoney(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string
debit[],int &s,int &i,int &c)
{
    int depositamount;
    cout<<" Enter the amount you want to deposit: ";
    cin>>depositamount;
    balance[c]+=depositamount;
    return true;
}

//Client will use this function to withdraw money.
```

```
bool withdrawmoney(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int &s,int &i,int &c)
```

```
{
    int withdrawamount;
    cout<<" Enter the amount you want to withdraw: ";
    cin>>withdrawamount;
    if(balance[c]>=withdrawamount)
    {
        balance[c]-=withdrawamount;
        return true;
    }
    else
        return false;
}
```

```
bool transfermoney(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],string &welfare,int &s,int &i,int &c)
```

```
{
    int transferamount;
    cout<<" Enter the amount you want to transfer: ";
    cin>>transferamount;
    cout<<" Enter the welfare to which yo want to transfer donations: ";
    cin>>welfare;
    if(balance[c]>=transferamount)
    {
        balance[c]-=transferamount;
        return true;
    }
    else
        return false;
}
```

```
}

//This function will issue Debit Card to client.

bool debitcard(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int
&s,int &i,int &c)
{
    string debitcard;
    int salary;
    cout<<" Which Debit Card you want to get(Visa , Master): ";
    cin>>debitcard;
    cout<<" Enter your monthly salary: ";
    cin>>salary;
    if(salary>=100000)
    {
        debit[c]=debitcard;
        return true;
    }
    else
    {
        return false;
    }
}

//This function will issue Loan to client.

bool getloan(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int
&s,int &i,int &c)
{
    int salary,loan1;
    cout<<" How much loan you you want to get: ";
    cin>>loan1;
    cout<<" Enter your salary: ";
    cin>>salary;
```



```
        if(salary>=100000)
        {
            loan[c]=loan1;
            return true;
        }
        else
        {
            return false;
        }
    }

//This function will deduct loan from client.

void payloan(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string debit[],int
&s,int &i,int &c)
{
    int pay;
    int loanpay;

    cout<<" Enter the amount of loan you want to pay: ";
    cin>>loanpay;
    if(balance[c]>=loanpay || balance[c]>=loan[c])
    {
        if(loan[c]==0)
            cout<<" You have no loan.";
        else if(loan[c]>loanpay)
        {
            balance[c]=balance[c]-loanpay;
            loan[c]=loan[c]-loanpay;
            cout<<" Your loan is successfully paid.";
        }
        else
```

```
{
    balance[c]=balance[c]-loan[c];
    cout<<"Your loan of "<<loan[c]<<" is successfully paid.";
    loan[c]=0;
}
}

else

    cout<<" Sorry! Your balance is insufficient. ";
}

//This function will submit complaints of client.

bool submitcomplaint(string name[],int password[],string role[],int balance[],int loan[],string complaints[],string
debit[],int &s,int &i,int &c)
{
    cout<<" Please write your Complaint: ";
    cin>>complaints[c];
    return true;
}

//This function will take amount,currency want to exchange and currency want to get from from and give desired
currency.

void exchangecurrency()
{
    float amount,exchange;
    string a,b;
    cout<<" Enter the amount you want to exchange: ";
    cin>>amount;
    cout<<" Enter your currency (Dollar, Pound, Rupee): ";
    cin>>a;
    cout<<" Enter your currency you want to get (Dollar, Pound, Rupee): ";
    cin>>b;
    if(a=="Rupee"&&b=="Dollar")
```

```
        exchange=amount/250;
        if(a=="Rupee"&&b=="Pound")
            exchange=amount/500;
        if(a=="Dollar"&&b=="Rupee")
            exchange=amount*250;
        if(a=="Dollar"&&b=="Pound")
            exchange=amount/2;
        if(a=="Pound"&&b=="Dollar")
            exchange=amount*2;
        if(a=="Pound"&&b=="Rupee")
            exchange=amount*500;
        cout<<" Recieve your "<<exchange<<" "<<b;
    }

//This function will display the total balance of clients and loan taken bt clients to manager.
void viewtotal(string name[],int password[],string role[],int balance[],int loan[],int &s,int &i,int &c)
{
    int totalb=0;
    int totall=0;
    for(int i=0;password[i]!=0;i++){
        totalb=totalb+balance[i];
        totall=totall+loan[i];}
    cout<<" Total Clients Balance in Bank is: "<<totalb;
    cout<<" \n Total Loan given to Clients is: "<<totall;
}

//This function will display the complaints of clients to manager.
bool viewcomplaints(string name[],int password[],string role[],string complaints[],int &s,int &i,int &c)
{
    int count=0;
```

```
    for(int i=0;name[i]!='\0';i++)
    {
        if(role[i]!="Manager"&&complaints[i]!='\0"&&name[i]!=" ")
        {
            cout<<" Complaint of Client "<<name[i]<<": "<<complaints[i]<<endl;
            count++;
        }
    }

    if(count==0)
        return true;
}

//Manager can open account of client with the help of this function.

bool openaccount(string name[],int password[],string role[],int &s,int &i,int &c)
{
    int x=0;
    while(true)
    {
        if(name[x]=="\0")
        {
            c=x;
            break;
        }
        x++;
    }

    cout<<" Enter Client Name: ";
    cin>>name[c];
    cout<<" Enter Client Password: ";
    cin>>password[c];
    role[c]="Client";
```

```
        return true;

        c++;

        i++;
    }

    //Manager can delete account of client with the help of this function.
    bool deleteaccount(string name[],int password[],string role[],int balance[],int &s,int &i,int &c)
    {
        int p=-1;
        string namen;
        int passwordn;
        cout<<" Enter Client Name: ";
        cin>>namen;
        cout<<" Enter Client Password: ";
        cin>>passwordn;
        for(int x=0;name[x]!='\0';x++)
        {
            if(name[x]==namen)
            {
                p=x;
                break;
            }
        }
        if(p!=-1)
        {
            cout<<" Account of Client "<<name[p]<<" is deleted. ";
            name[p]=" ";
            balance[p]=0;
            return false;
        }
    }
}
```

```
    }
    else
    {
        return true;
    }
}

//This function will assist manager in hiring staff.
bool hirestaff(string staff[],string des[],int &s,int &i,int &c)
{
    cout<<" Enter Employee Name: ";
    cin>>staff[s];
    cout<<" Enter Employee Designation: ";
    cin>>des[s];
    return true;
    s++;
}

//This function will assist manager in expelling staff.
bool expelstaff(string staff[],string des[],int &s,int &i,int &c)
{
    string n,d;
    cout<<" Enter Employee Name: ";
    cin>>n;
    cout<<" Enter Employee Designation: ";
    cin>>d;
    int x=-1;
    for(int i=0;i<=s;i++)
    {
        if(staff[i]==n&&des[i]==d)
        {
```

```
        x=i;
        break;
    }
}
if(x!=-1)
{
    cout<<" Employee "<<staff[x]<<" is expelled. ";
    staff[x]=" ";
    des[x]==" ";
    return false;
}
else
    return true;
}

//This function will display manager all staff of bank.
bool viewstaff(string staff[],string des[],int &s,int &i,int &c)
{
    int count=0;
    for(int i=0;i<=s;i++)
    {
        if(staff[i]!=" "&&staff[i]!="\0")
        {
            count++;
        }
    }
    if(count==0)
        return true;
    else{
```

```
        cout<<" Employee Name\t\tDesignation \n";
    for(int i=0;i<=s;i++)
    {
        if(staff[i]!=" ")
        cout<<" "<<staff[i]<<"\t\t"<<des[i]<<endl;
    }
}

}

//This function will assist manager to block the debit card of client.
bool blockdebit(string name[],int password[],string role[],string debit[],int &s,int &i,int &c)
{
    int count=0;
    int p=-1;
    string namev;
    cout<<" Enter Client name: ";
    cin>>namev;
    for(int x=0;name[x]!="\0";x++)
    {
        if(name[x]==namev)
        count++;
    }
    if(count==0){
        return true;}
    else{
        for(int y=0;name[y]!="\0";y++)
        {
            if(name[y]==namev&&debit[y]!="N/A")
            p=y;
        }
    }
}
```



```
    }  
    if(p!=-1)  
    {  
        debit[p]="N/A";  
        cout<<" The debit card of Client "<<name[p]<<" is blocked."  
    }  
    else{  
        cout<<" Sorry! The Client has no debit card.;"  
        return false;  
    }  
}  
  
//This function will exit manager and client from their menus and place them in sign in or sign up options.  
bool exit()  
{  
    system("cls");  
    return true;  
}
```

- **Weakness in the Business Application**

- My Business Application cannot permanently store data. When my Bank Management System is closed and reopened the data of all clients is lost.
- In every function I have passed a long list of parameters due to which the code of my business application become difficult to read and understand.

- **Future Directions**

- In next semester, I want to make business application that can permanently store data i.e even if the Bank Management System is closed the data of clients is saved in the system and when it is reopened their data remain the same. My Bank Management System can keep record of all banking transactions.