**<<CS-1001L>>**

**<<Programming Fundamentals>>**

**<<Bank Management System >>**

**Project Report By**

**<<Shazaib Shiraz>>**

**<<CS231130>>**

**Group Members:**

<<Shazaib Shiraz>> <<CS231130>>

<<Karan Kumar>> <<CS231145>>

<<Aliyan Tajddin>> <<CS231141>>

<<Abdul Rehman>> <<CS231140>>

<<Arman Siraj>> <<CS231126>>

**Table of Contents**

* **Introduction**

The Customer Account Banking Management System is a software application designed to facilitate banking operations for customers. Developed by Shazaib Shiraz, Abdul Rehman, Arman, Karan, and Aliyan, the system provides a user-friendly interface for account creation, deposit, withdrawal, and balance inquiries.

* **Scope**

The scope of the Customer Account Banking Management System encompasses basic banking functionalities such as account creation, deposit, withdrawal, and balance inquiries. It caters to individuals seeking efficient management of their banking activities. The system targets small to medium-sized banks or financial institutions aiming to streamline customer transactions and enhance operational efficiency.

**3. Features**

1**. Account Creation**: Users can create new accounts by providing essential details such as name, contact number, date of birth, CNIC (Computerized National Identity Card) number, and an initial deposit amount and also requires password for security purpose.

2. **Deposit**: Account holders can deposit funds into their accounts, But firstly the user have to enter the password which he/she has entered while account creation and then they can access to credit the amount to increase their balance. The system updates the account balance accordingly and maintains transaction records.

3. **Withdrawal**: Customers can withdraw funds from their accounts, provided they have sufficient balance. The system validates withdrawal requests against available funds and updates the account balance post-withdrawal.

4. **Balance Inquiry**: Users can check their account balances at any time. The system retrieves the current balance from the database and displays it to the user upon request.

5. **User-Friendly Interface**: The system features a simple and intuitive user interface, making it easy for customers to navigate through various functionalities.

6. **Error Handling:** The system incorporates error handling mechanisms to validate user inputs and provide appropriate error messages in case of invalid inputs or insufficient funds.

* **Requirements (Functional/Non-Functional/Hardware/Software)**

**Functional Requirements:**

**Account Creation**: Users must provide essential details for account creation, including name, contact number, date of birth, CNIC, and initial deposit amount.

**Deposit**: The system should allow users to deposit funds into their accounts.

**Withdrawal**: Users should be able to withdraw funds from their accounts, ensuring that the withdrawal amount does not exceed the available balance.

**Balance Inquiry**: The system should provide users with the ability to check their account balances at any time.

**Non-Functional Requirements**:

User Interface: The system should have a user-friendly interface, enhancing user experience and accessibility.

Security: The system must ensure the security and confidentiality of user data and transactions.

Performance: The system should be responsive and capable of handling multiple user requests simultaneously.

Reliability: The system should be reliable, with minimal downtime and accurate transaction processing.

**Hardware Requirements**:

The system can run on standard desktop or laptop computers.

It requires sufficient storage capacity to store user data and transaction records.

It should be compatible with common peripherals such as printers for generating transaction receipts.

**Software Requirements:**

The system requires an operating system (e.g., Windows, Linux) to run the application.

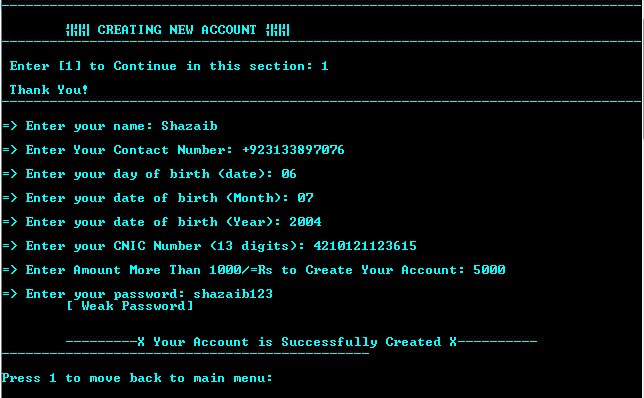
It may utilize programming languages such as C for application development.

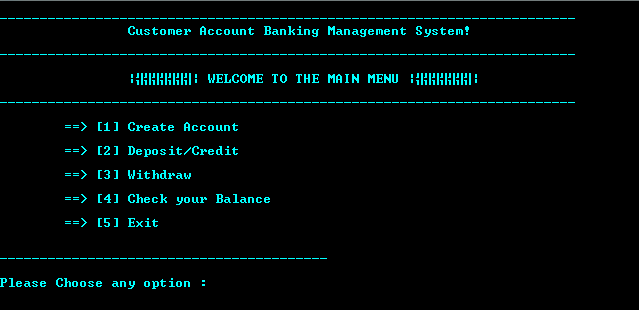
Database management systems (e.g., SQLite, MySQL) may be used to store and manage user data and transaction records.

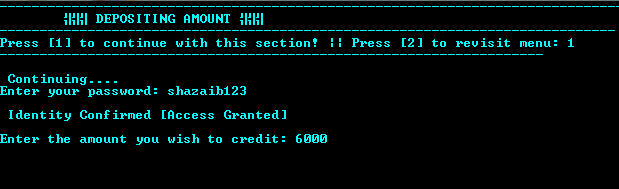
Text editors or integrated development environments (IDEs) are needed for coding and debugging purposes.

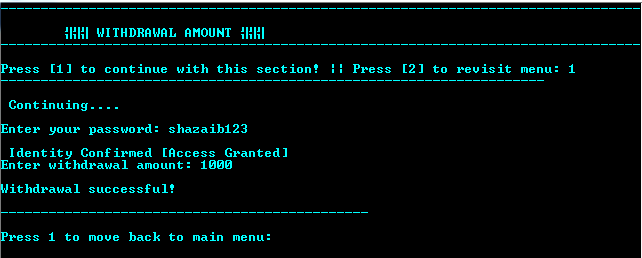
By fulfilling these requirements, the Customer Account Banking Management System aims to provide a seamless banking experience for customers while ensuring operational efficiency and data integrity.

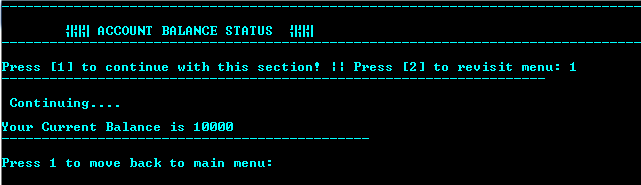
* **Screen Shots**

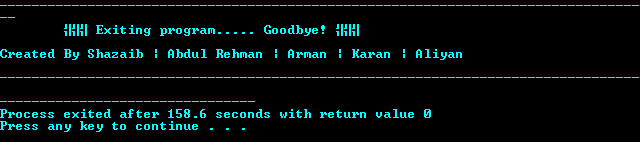




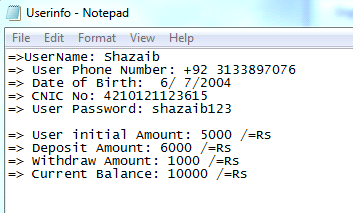








**The program creates Text file as userinfo.txt to Store User’s complete information**



**Flow Chart**

Main Menu Display

User input

Switch Case (Choice)

|

| 🡪 Case 1 (Create Account)

| |

| V

| Call create account Function

| |

| V

| Write Account Details to File

| |

| V

| Continue or Back to Menu

|

|🡪 Case 2 (Deposit)

| |

| V

| Call deposit Function

| |

| V

| Check if User has Account

| |

| V

| Prompt for Password and Validate

| |

| V

| Credit Account and Update File

| |

| V

| Continue or Back to Menu

|

|🡪 Case 3 (Withdraw)

| |

| V

| Call withdraw Function

| |

| V

| Check if User has Account

| |

| V

| Prompt for Password and Validate

| |

| V

| Process Withdrawal and Update File

| |

| V

| Continue or Back to Menu

|

|🡪 Case 4 (Check Balance)

| |

| V

| Call check balance Function

| |

| V

| Check if User has Account

| |

| V

| Display Current Balance and Update File

| |

| V

| Continue or Back to Menu

|

|🡪 Case 5 (Exit)

|

V

Call quit program Function

|

V

Display Goodbye Message and Exit

|

V

End

**7. Code**

#include <stdio.h>

#include <stdlib.h>

#include <string.h>

#include <time.h>

void backtomenu();

void menu();

void createaccount(); //Function to Create Account

void deposit(); //Function to Deposit

void withdraw(); //Function to Withdraw amount

void checkbalance(); //Function to check the balance

void quitprogram(); //exit function

void Continue(); //checks whether user wanted to access this section

void option\_check(); //verfies your existence in the first section which allows you to access other sections

//Some Global variable Declarations

int balance, initial\_amount, credit\_amount, Withdraw\_amount,check=0; //Everything an account does

char pass[20],cpass[20]; //password & confirm password strings

int confirmation,confirmation2; //password verifications

int length; //length of password which determines its strength

int choice=0; //switch case variable

int main ()

{

do{

menu(); //function to print main menu

printf("\nPlease Choose any option : ");

scanf("%d",&choice);

switch(choice){

case 1:

createaccount(); //Function to Create Account

break;

case 2:

deposit(); //Function to Deposit

break;

case 3:

withdraw(); //Function to Withdraw amount

break;

case 4:

checkbalance(); //Function to check the balance

break;

case 5:

quitprogram(); //exit function

break;

default:

printf("Invalid Menu option");

break;

}

} while (1); //To run loop multiple times

}

void menu(){

system("cls"); // Built-in Function To Clear Screen

system ("color B");

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

printf("\t\tCustomer Account Banking Management System!\n\t");

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

printf("\n\t\t|\xB9\xB9\xB9\xB9\xB9\xB9\xB9| WELCOME TO THE MAIN MENU |\xB9\xB9\xB9\xB9\xB9\xB9\xB9|\n");

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

printf("\t==> [1] Create Account\n\n");

printf("\t==> [2] Deposit/Credit \n\n");

printf("\t==> [3] Withdraw\n\n");

printf("\t==> [4] Check your Balance\n\n");

printf("\t==> [5] Exit\n\n");

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

}

void createaccount(){ //Function to Create Account

system("cls"); // Built-in Function To Clear Screen

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

printf("\t\xB9\xB9\xB9 CREATING NEW ACCOUNT \xB9\xB9\xB9\n");

printf("--------------------------------------------------------------------------------");

//Continue function wasnt working here [Reason: Unknown]

char name[20], phone\_number[15],account\_type,CNIC[15];

int dd,mm,yy;

option\_check(); //This is what allows you to access other sections

printf("\n=> Enter your name: ");

scanf("%s",&name);

// Validate phone number

do {

printf("\n=> Enter Your Contact Number: +92");

scanf("%s", &phone\_number);

if (strlen(phone\_number) != 10) {

printf("Invalid Contact number <<<RETRY>>>\n");

} else {

break;

}

} while (1); //To run loop for as long as the phone number limit is true

while(1)

{

printf("\n=> Enter your day of birth (date): ");

scanf("%2d", &dd);

printf("\n=> Enter your date of birth (Month): ");

scanf("%2d",&mm);

printf("\n=> Enter your date of birth (Year): ");

scanf("%4d",&yy);

if ((dd < 1 || dd > 31) || (mm < 1 || mm > 12) || (yy < 1900 || yy > 2024)) {

printf("Invalid date format <<<RETRY>>>\n");

} else {

break;

}

}

// Validate CNIC

do {

printf("\n=> Enter your CNIC Number (13 digits): ");

scanf("%s", &CNIC);

if (strlen(CNIC) != 13) {

printf("\nIncorrect CNIC <<<RETRY>>>\n");

} else {

break;

}

} while (1); //To run loop multiple times

// Validate initial amount

do {

printf("\n=> Enter Amount More Than 1000/=Rs to Create Your Account: ");

scanf("%d",&initial\_amount);

if (initial\_amount < 1000) {

printf("\nInvalid Amount <<<RETRY>>>\n");

} else {

break;

}

} while (1); //To run loop multiple times

balance +=initial\_amount;

printf("\n=> Enter your password: ");

scanf("%s",&pass);

length=strlen(pass);

if(length>10){

printf("\t[Strong Password]"); }

else {

printf("\t[ Weak Password]");

}

printf("\n\n\n\t---------X Your Account is Successfully Created X----------");

FILE \*fp; //Filing to create file to store user account info

fp = fopen("Userinfo.txt", "w");

if (fp == NULL) {

printf("\nError opening file!\n");

exit(1);

}

fprintf(fp, "=>UserName: %s \n=> User Phone Number: +92 %s \n=> Date of Birth: %2d/%2d/%4d \n=> CNIC No: %s \n=> User Password: %s\n\n=> User initial Amount: %d /=Rs", name, phone\_number, dd, mm, yy, CNIC,pass,initial\_amount);

fclose(fp);

backtomenu();

}

void deposit(){ //Function to Deposit

system("cls"); // Built-in Function To Clear Screen

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

printf("\t\xB9\xB9\xB9 DEPOSITING AMOUNT \xB9\xB9\xB9\n");

printf("-----------------------------------------------------------------------------");

if(check==1) //this checks if you already have an account or not

{

Continue();

}

else

{

printf("\n Go Create an Account First");

sleep(2);

main();

}

b:

printf("Enter your password: ");

scanf("%s",&cpass);

confirmation=strcmp(pass,cpass);

confirmation2=strcmp(cpass,pass);

if(confirmation==confirmation2)

{

printf("\n Identity Confirmed [Access Granted]\n");

goto a;

}

else

{

printf("\n Identiy Uncomfirmed [Access Denied]\n");

goto b;

}

a:

printf("\nEnter the amount you wish to credit: ");

scanf("%d", &credit\_amount);

balance += credit\_amount;

FILE \*fp; //Filing to append the credit amount in user account info

fp = fopen("Userinfo.txt", "a");

if (fp == NULL) {

printf("\nError opening file!\n");

exit(1);

}

fprintf(fp, "\n=> Deposit Amount: %d /=Rs",credit\_amount);

fclose(fp);

}

void withdraw(){ //Function to Withdraw amount

system("cls"); // Built-in Function To Clear Screen

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

printf("\t\xB9\xB9\xB9 WITHDRAWAL AMOUNT \xB9\xB9\xB9\n");

printf("--------------------------------------------------------------------------------");

if(check==1) //this checks if you already have an account or not

{

Continue();

}

else

{

printf("\n Go Create an Account First");

sleep(2);

main();

}

b:

printf("\nEnter your password: ");

scanf("%s",&cpass);

confirmation=strcmp(pass,cpass);

confirmation2=strcmp(cpass,pass);

if(confirmation==confirmation2)

{

printf("\n Identity Confirmed [Access Granted]");

goto a;

}

else

{

printf("\n Identiy Uncomfirmed [Access Denied]");

goto b;

}

a:

printf("\nEnter withdrawal amount: ");

scanf("%d", &Withdraw\_amount);

if (Withdraw\_amount > balance) {

printf("\nInsufficient balance!\n");

} else {

balance -= Withdraw\_amount;

printf("\nWithdrawal successful!\n");

FILE \*fp; //Filing to append the Withdraw amount in user account info

fp = fopen("Userinfo.txt", "a");

if (fp == NULL) {

printf("\nError opening file!\n");

exit(1);

}

fprintf(fp, "\n=> Withdraw Amount: %d /=Rs",Withdraw\_amount);

fclose(fp);

}

backtomenu();

}

void checkbalance(){ //Function to check the balance

system("cls"); // Built-in Function To Clear Screen

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

printf("\t\xB9\xB9\xB9 ACCOUNT BALANCE STATUS \xB9\xB9\xB9\n");

printf("--------------------------------------------------------------------------------");

if(check==1)

{

Continue();

goto b;

}

else

{

printf("\n Go Create an Account First");

sleep(2);

main();

}

b:

balance = initial\_amount + credit\_amount - Withdraw\_amount;

printf("\nYour Current Balance is %d", balance);

FILE \*fp; //Filing to append the current balance in user account info

fp = fopen("Userinfo.txt", "a");

if (fp == NULL) {

printf("\nError opening file!\n");

exit(1);

}

fprintf(fp, "\n=> Current Balance: %d /=Rs",balance);

fclose(fp);

backtomenu();

}

void quitprogram(){ //function to Exit program

system("cls"); // Built-in Function To Clear Screen

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

printf("\t\xB9\xB9\xB9 Exiting program..... Goodbye! \xB9\xB9\xB9\n");

printf("\nCreated By Shazaib | Abdul Rehman | Arman | Karan | Aliyan \n");

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

exit(0); // Exit program successfully};

}

void backtomenu(){

int i;

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

printf ("\nPress 1 to move back to main menu: ");

scanf("%d",&i);

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

if(i==1){

menu();

}

else

{

printf("\nInvalid Key is being pressed!\n");

}

}

void Continue(){

int i;

printf("\nPress [1] to continue with this section! || Press [2] to revisit menu: ");

scanf("%d",&i);

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

if(i==2){

menu();

}

else if(i==1)

{

printf("\n Continuing....\n");

}

}

void option\_check()

{

do{

printf("\n Enter [1] to Continue in this section: ");

scanf("%d",&check); //without this check you will not be able to access other options

if(check==1)

{

printf("\n Thank You!\n");

printf("--------------------------------------------------------------------------------");

break;

}

else

{

menu();

}

}while(1);

}\

**Project breakdown table**

|  |  |  |  |
| --- | --- | --- | --- |
| **S #** | **Reg. #** | **Name** | **Tasks** |
| 1. | **<CS231130>** | **<Shazaib Shiraz>** | * Complete User interface and designs including program color scheme (0B) * Complete file handling of the program * Creating functions for account creation. * Creating logics to get appropriate outputs. * Fix all the bugs and errors in program. |
| 2. | **<CS231145>** | **<Karan Kumar>** | * Creating functions for deposit and withdrawal. * Taking input and output from users * Tackling with user data |
| 3. | **<CS231141>** | **<Aliyan Tajddin>** | * Creating functions for current balance and exit program. * Taking input and output from users * Tackling with user data |
| 4. | **<CS231140>** | **<Abdul Rehman>** | * Complete program validations and limitations for user, including an option that checks whether user has created an account which then allows user to access to other sections in program. * Applying password for security * Creating functions for program. * Creating continue function for user convenience. * Fix all the bugs and errors in program |
| 5. | **<CS231126>** | **<Arman Siraj>** | * Creating functions for return to menu. * Taking input and output from users * Tackling with user data |