Banking Management System

Complete features:

The full Code-

```
#include <stdio.h>
#include <string.h>
#include <time.h>
#include <conio.h>
#include <stdlib.h>
#define MAX_ACCOUNT_NUMBER 9999
#define MIN_ACCOUNT_NUMBER 1000
struct user_account {
   char usr_name[30] ;
   int id;
   float balance;
   char password[8];

} user, check;
struct Admin {
   int id_admin;
   char pass_admin[8] ;

} admin,ad_check;
char* timeDate() {
   static char buffer[100];

   time_t now = time(NULL);

   struct tm *local = localtime(&now);


   snprintf(buffer, sizeof(buffer), "- %02d:%02d:%02d - %02d/%02d/%d",local->tm_hour,
local->tm_min, local->tm_sec,local->tm_mday, local->tm_mon + 1, local->tm_year + 1900);


   return buffer;

}

void delay(int number_of_seconds){

   int milli_seconds = 1000 * number_of_seconds;

   clock_t start_time = clock();

   while (clock() < start_time + milli_seconds);

}

int accountExists(int newID) {

   // system("color a0");

   struct user_account user;

   FILE *info;


   info = fopen("info.txt", "r");

   if (info == NULL) {

       printf("\n\t\tCould not open file.\n");

       return 0;

   }


   while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, user.password, &user.balance)
!= EOF) {

       if (newID == user.id) {
           fclose(info);
           return 1;
       }
   }
   fclose(info);
   return 0;
}
int generateRandomAccountNumber() {
   int accountNumber;
   do {
       accountNumber = (rand() % (MAX_ACCOUNT_NUMBER - MIN_ACCOUNT_NUMBER + 1)) + MIN_ACCOUNT_NUMBER;
   } while (accountExists(accountNumber));
   return accountNumber;
}
void createAccount() {
   FILE *info;
   struct user_account new_user;
   info = fopen("info.txt", "a");

   if (info == NULL) {
       printf("Error opening file!\n");
       return;
   }

   printf("\n\t       Add record      \n");
   printf("\tEnter name: ");
   getchar();
   scanf("%s", new_user.usr_name);

   new_user.id = generateRandomAccountNumber();
   printf("\n\tGenerated Account Number: %d\n", new_user.id);

   while (1) {
       printf("\tEnter Password, max 8 characters: ");
       scanf("%s", new_user.password);
       int len = strlen(new_user.password);

       if (len <= 8) {
           new_user.balance = 0;
           fprintf(info, "%s %d %s %.2f\n", new_user.usr_name, new_user.id,
new_user.password, new_user.balance);

           printf("\n\t\tAccount Created Successfully\n");
           break;
       } else {
           printf("\n\t\tPassword exceeded maximum length! Please set within 8 characters.\n");
       }
   }

   fclose(info);
   delay(2);
}
void depositMoney(int id,float balance){
   char trans_type[20] = "DIPOSIT-";

   FILE *info;

   FILE *newinfo;

   FILE *trans;

   trans = fopen("transaction.txt","a");

   info = fopen("info.txt","r");

   newinfo = fopen("newinfo.txt","w");

               while(fscanf(info,"%s %d %s %f",user.usr_name,&user.id,&user.password,&user.balance)!= EOF){

                   if(id == user.id){

                       user.balance += balance;

                       fprintf(trans,"%s %d %s %.2f %s\n",user.usr_name,user.id,trans_type,balance,timeDate());

                   }

                   fprintf(newinfo,"%s %d %s %.2f\n",user.usr_name,user.id,user.password,user.balance);


               }

   fclose(trans);

   fclose(newinfo);

   fclose(info);


   remove("info.txt");

   rename("newinfo.txt", "info.txt");


}

void withdrawMoney(int id,double balance){

   struct user_account user;
   char trans_type[20] = "WITHDRAW";
   FILE *info,*newinfo,*trans;
   trans = fopen("transaction.txt","a");
   info = fopen("info.txt","r");
   newinfo = fopen("newinfo.txt","w");

   while(fscanf(info,"%s %d %s %f",user.usr_name,&user.id,&user.password,&user.balance)!=
EOF){
        if(id == user.id){
            if(balance > user.balance){
               printf("\n\t\t\tInsufficient Balance!!\n\n");

            }else{
               user.balance -= balance;
               fprintf(trans,"%s %s %d %s %.2f
\n",timeDate(),user.usr_name,user.id,trans_type,balance);
            }
        }
        fprintf(newinfo,"%s %d %s %.2f\n",user.usr_name,user.id,user.password,user.balance);
   }
   fclose(trans);
   fclose(newinfo);
   fclose(info);

   remove("info.txt");
   rename("newinfo.txt","info.txt");
}
void sendMoney(int sen_id, int rec_id, float balance) {
}



   struct user_account user;
   FILE *info;
   info = fopen("info.txt", "r");
   while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, &user.password,
&user.balance) != EOF) {
        if(sen_id == user.id){
            fclose(info);
            if(balance > user.balance){
               printf("\n\t\t\tInsufficient Balance!!\n\n");
            }else{
               if(accountExists(rec_id)){
                   withdrawMoney(sen_id,balance);
                   depositMoney(rec_id,balance);
                   printf("\n\t\t\tSent money successfully");


               }
               else{
                   printf("\n\t\tRecipient account not found.\n\n");
               }
            }


        }


   }


}
```

```
void checkBalance(int id){

    struct user_account user;
    FILE *info;
    info = fopen("info.txt","r");
    printf("\n\t\tCheck Balance\t\n");

    while(fscanf(info,"%s %d %s %f",user.usr_name,&user.id,&user.password,&user.balance)!=
EOF){
        if(id == user.id){
            printf("\n\t\t%s,Your current Balance is : %.2f\n\n\n",user.usr_name,user.balance);
            break;
        }
    }

    fclose(info);
}
void accountDetails(int id) {
    struct user_account user;
    FILE *info, *trans;
    info = fopen("info.txt", "r");
    if(accountExists(id) == 0){
        printf("\n\n\t\tAccount Doesnot Exist....\n");
    }
    while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, &user.password,
&user.balance) != EOF) {
        if (id == user.id) {
            printf("\n\n\t\t  Account Details :\n\n\t\t\t  Name          : %s\n\t\t\t  Account Number
: %d\n\t\t\t  Account Balance: %.2f\n",user.usr_name,user.id,user.balance);
            break;
        }
    }
    fclose(info);

    char time[50], name[20];
    float balance;
    int p_id, flag = 1;
    char trans_type[20];

    trans = fopen("transaction.txt", "r");
    printf("\n\n\t\t  Transaction Details:\n\n");
    while (fscanf(trans, "%s %d %s %f", name, &p_id, trans_type, &balance) != EOF) {
        fgets(time, sizeof(time), trans);
        time[strcspn(time, "\n")] = 0;
        if (id == p_id) {
            printf("\t\t%-10s %.2f at %s\n", trans_type, balance, time);
            flag = 0;
        }
    }
    if(flag){
        printf("\n\t\t\tNo transaction yet.\n");
    }
    fclose(trans);
    printf("\n\t Press Enter to exit.");

    getch();

    system("cls");

}

void login(struct user_account user){


        FILE *info;

        struct user_account check;

        int m,id;

        float balance;


        login_menu:

            //fclose(info);

            system("cls");

            printf("       \n\t\t\t\tBanking Management System     \n\n");

            printf("\t\t\t1. Deposit Money\n");

            printf("\t\t\t2. Withdraw Money\n");

            printf("\t\t\t3. Send Money\n");

            printf("\t\t\t4. Check Balance\n");

            printf("\t\t\t5. Account Details\n");

            printf("\t\t\t6. Logout\n");

            printf("\n\tEnter your choice: ");

            scanf("%d", &m);


            switch(m){


            case 1:
```

```
    printf("\t\tEnter money to deposit : ");

    scanf("%f",&balance);

    depositMoney(user.id,balance);

    printf("\n\t\tDiposit successfull.\n\n");

    delay(2);

    goto login_menu;

    break;


case 2:

    printf("\n\n\tWithdraw money :\n");

    printf("\n\tEnter money to withdraw : ");

    scanf("%f",&balance);

    withdrawMoney(user.id,balance);

    printf("\n\n\t\tSuccessfully Withdrawed money !\n");

    delay(2);

    goto login_menu;

    break;


case 3:

    printf("\n\n\tSend money\n");

    printf("\t\tEnter Account number : ");

    scanf("%d",&id);

    printf("\t\tEnter money to send : ");

    scanf("%f",&balance);


    sendMoney(user.id, id, balance);
```

```
  delay(2);
  goto login_menu;
  break;

case 4:
  checkBalance(user.id);
  delay(2);
  goto login_menu;
  break;

case 5:
  accountDetails(user.id);
  goto login_menu;
  break;

case 6:
  printf("\n\t\t\t\tExiting...\n");
  delay(2);
  user_menu();
  break;

default:
  printf("\tInvalid choice! Please try again.\n");
  delay(1);
  goto login_menu ;
  break;
}
```

```
}
void user_menu(){

    int m;
    FILE *info;
    struct user_account user,check;

        system("cls");
        printf("      \n\t\t\t\tBanking Management System     \n\n");
        printf("\n\t\t\t1. Login");
        printf("\n\t\t\t2. Create Account\n");
        printf("\t\t\t3. Exit \n");

      printf("\n\tEnter your choice: ");
      scanf("%d", &m);

      switch (m) {
      case 1:
      usr_menu:
        info = fopen("info.txt","r");
        printf("\n\t\tEnter Account number: ");
        scanf("%d",&check.id);
        //getchar();
        fflush(stdin);
        printf("\n\t\tEnter Password :");
        scanf("%s",check.password);
        while(fscanf(info,"%s %d %s
%f",user.usr_name,&user.id,&user.password,&user.balance)!= EOF){
          if( (check.id == user.id) && (strcmp(check.password, user.password) == 0)){
            printf("\n\t\tLogin successful! Welcome, %s.\n\n", user.usr_name);
            fclose(info);
            delay(2);
            login(user);
            //goto user_menu;
          }
        }
        printf("\n\t\tLogin Failed! Re-enter ID and Password.\n\n");
        fclose(info);
        delay(2);
        goto usr_menu;

        break;

      case 2:
        createAccount(user);
        user_menu();
        //login(user);
        break;

      case 3:
        printf("\n\t\tExiting ....\n\n");
        delay(2);
```

```
                system("cls");
                main();
                break;

            default:
                printf("\n\t\t\tInvalid choice! Please try again.\n");
                delay(2);
                user_menu();
            }
}
void updateInfo() {
    struct user_account user, check;
    FILE *info, *newinfo;
    int flag = 1;
    info = fopen("info.txt", "r");
    newinfo = fopen("newinfo.txt", "w");

    printf("\n\nEnter the account number of the user whose info you want to change: ");
    scanf("%d", &check.id);


    while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, user.password, &user.balance) != EOF) {
        if (check.id == user.id) {
            flag = 0;
            printf("\n\n\t\tEnter account holder's correct name : ");
            getchar();
            fgets(check.usr_name, sizeof(check.usr_name), stdin);
            if (check.usr_name[strlen(check.usr_name) - 1] == '\n') {
                check.usr_name[strlen(check.usr_name) - 1] = '\0';
            }
            strcpy(user.usr_name, check.usr_name);

            printf("\n\t\tName changed successfully.\n\n");
        }
        fprintf(newinfo, "%s %d %s %.2f\n", user.usr_name, user.id, user.password,
user.balance);

    }
    if(flag){
        printf("\n\t\tAccount doesn't exists....\n\n");
        delay(2);
    }

    fclose(newinfo);
    fclose(info);

    remove("info.txt");
    rename("newinfo.txt", "info.txt");

}
void deleteAccount() {
    struct user_account user;
    int id;
    FILE *info, *newinfo;
    info = fopen("info.txt", "r");
    newinfo = fopen("newinfo.txt", "w");

    printf("\n\tEnter User Account number : ");
    scanf("%d", &id);

    int accountFound = 1;

    while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, user.password, &user.balance) != EOF) {
        if (id == user.id) {
            accountFound = 0;
            continue;
        }
        fprintf(newinfo, "%s %d %s %.2f\n", user.usr_name, user.id, user.password, user.balance);
    }

    if(accountFound){
        printf("\n\n\t\tAccount doesn't Exist.\n");
    }
    else{
        printf("\n\n\t\tSuccessfully Deleted Account!\n");
    }
    fclose(newinfo);
    fclose(info);

    //if (accountFound) {
        remove("info.txt");
        rename("newinfo.txt", "info.txt");

   // } else {
        //printf("\n\t\t Account no. %d is not found.\n", id);
        remove("newinfo.txt");
    //}
    delay(2);

}
void report(){
    FILE *trans;
    char trans_type[20],time[70],name[20];
    float balance;
    int id;
    trans = fopen("transaction.txt","r");
    printf("\n\t\t Transaction report :\n");
    printf("\n\tName\tAc.No \tTrans.Type   \tMoney       Time\n\n");
    while(fscanf(trans,"%s %d %s %f",name,&id ,trans_type, &balance) != EOF){
        fgets(time, sizeof(time), trans);
        time[strcspn(time, "\n")] = 0;
        printf("\t%s   %d       %s       %.2f      %s\n",name,id,trans_type, balance, time);
    }
    fclose(trans);
    printf("\n\t\tEnter to exit.");
    getch();

}
void adminMenu(){

    int m,id;
    while(1){

        printf("      \n\t\t\t\tBanking Management System     \n\n");
        printf("\t\t\tAdmin Menu \n\n");
        printf("\t\t1. See User Information\n");
        printf("\t\t2. Update User Information\n");
        printf("\t\t3. Delete User account\n");
        printf("\t\t4. Transaction Report \n");
        printf("\t\t5. Logout \n");
        printf("\nEnter your choice: ");
        scanf("%d", &m);
        switch(m){
            case 1:

                printf("\n\t\tEnter account number : ");
                scanf("%d",&id);
                if(accountExists(id)){
                    accountDetails(id);
                }
                else{
                    printf("\n\t\tAccount doesn't Exist.Press Enter to proceed.");
                }
                getchar();
                system("cls");
                break;
            case 2:
                updateInfo();
                delay(2);
                system("cls");
                break;
            case 3:
                deleteAccount();
                system("cls");
                break;

            case 4:
                report();



                system("cls");
                break;

            case 5:
                printf("\n\t\tExiting ...\n\n");
                delay(2);
```

```
                system("cls");
                main();
                break;
            default:
                printf("\t\tInvalid choice! Please try again.\n");
                delay(2);
                system("cls");
                adminMenu();
                break;
        }
        system("cls");
    }
}
int main() {
    system("color e0");
    FILE *info;
    struct user_account user, check;
    struct Admin admin,ad_check;
    int n,m;

    while (1) {
        printf("      \n\t\t\t\tBanking Management System     \n\n");
        printf("\t\t\t1.User\n\t\t\t2.Admin\n");
        printf("\n\tEnter your choice: ");
        scanf("%d", &n);

        switch(n){
```

```
        case 1:
            user_menu();
            break;

        case 2:
            printf("\n\tEnter Admin passsword: ");
            scanf("%s", &ad_check.pass_admin);
            if((strcmp(ad_check.pass_admin , "pass") == 0)){
                system("cls");
                adminMenu();
            }
            else{
                printf("\t\tInvalid Password,try again.");
                getchar();
                system("cls");
            }
            break;

        default :
            printf("\n\n\t\t\t\tInvalid option.\n\t\t\t\t\tTry again.....\n");
            break;


    }
    //system("cls");
}
return 0;
```

```
}
```

Project Features

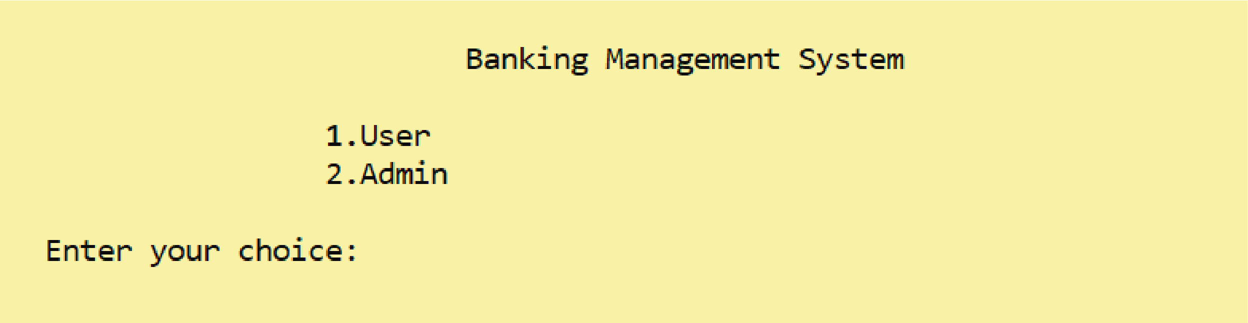
The Banking Management System will include the following core features:

Front page---

The function code:--

```
 int main() {
    system("color e0");
    FILE *info;
    struct user_account user, check;
    struct Admin admin,ad_check;
    int n,m;
    while (1) {
        printf("      \n\t\t\t\tBanking Management System     \n\n");
        printf("\t\t\t1.User\n\t\t\t2.Admin\n");
        printf("\n\tEnter your choice: ");
        scanf("%d", &n);
        switch(n){
            case 1:
                user_menu();
                break;
            case 2:
                printf("\n\tEnter Admin passsword: ");
                scanf("%s", &ad_check.pass_admin);
                if((strcmp(ad_check.pass_admin , "pass") == 0)){
                    system("cls");
                    adminMenu();
        }
       else{
          printf("\t\tInvalid Password,try again.");
          getchar();
          system("cls");
        }
        break;

      default :
        printf("\n\n\t\t\t\tInvalid option.\n\t\t\t\t\tTry again.....\n");
        break;

    }
    //system("cls");
  }
  return 0;
}
```

Output:



```
                                  Banking Management System

                        1.User
                        2.Admin

   Enter your choice:
```

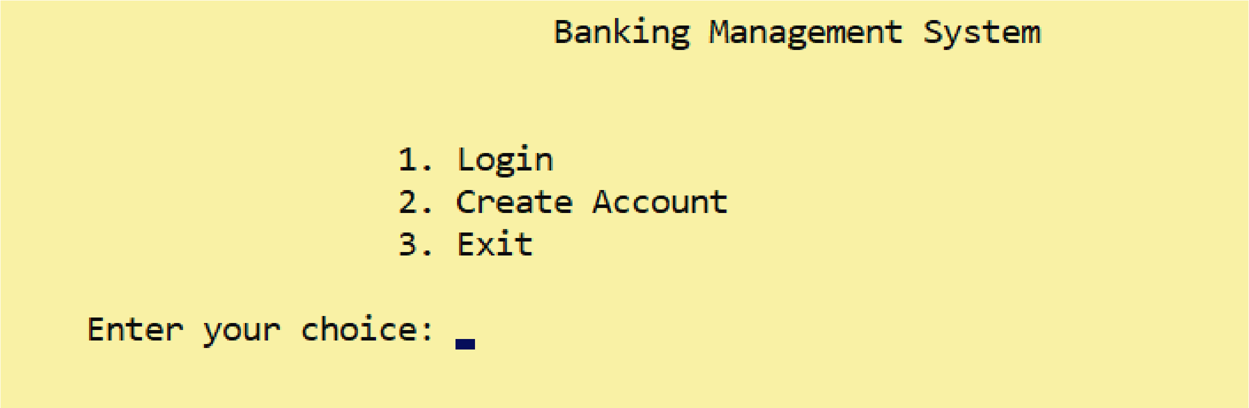
Users menu----

The code:-

```
void user_menu(){
    int m;
    FILE *info;
    struct user_account user,check;
        system("cls");
        printf("      \n\t\t\t\tBanking Management System     \n\n");
        printf("\n\t\t\t1. Login");
        printf("\n\t\t\t2. Create Account\n");
        printf("\t\t\t3. Exit \n");
      printf("\n\tEnter your choice: ");
      scanf("%d", &m);
```

```
         switch (m) {
        case 1:
         usr_menu:
            info = fopen("info.txt","r");
            printf("\n\t\tEnter Account number: ");
            scanf("%d",&check.id);
            //getchar();
            fflush(stdin);
            printf("\n\t\tEnter Password :");
            scanf("%s",check.password);
            while(fscanf(info,"%s %d %s
%f",user.usr_name,&user.id,&user.password,&user.balance)!= EOF){
               if( (check.id == user.id) && (strcmp(check.password, user.password)
== 0)){
                  printf("\n\t\tLogin successful! Welcome, %s.\n\n",
user.usr_name);
                  fclose(info);
                  delay(2);
                  login(user);
                  //goto user_menu;
               }
            }
            printf("\n\t\tLogin Failed! Re-enter ID and Password.\n\n");
            fclose(info);
            delay(2);
            goto usr_menu
            break;
         case 2:
            createAccount(user);
            user_menu();
            //login(user);
            break;
         case 3:
            printf("\n\t\tExiting ....\n\n");
            delay(2);
            system("cls");
            main();
            break;
         default:
            printf("\n\t\t\tInvalid choice! Please try again.\n");
            delay(2);
            user_menu();
         }
}
```

```
}
```



```
                          Banking Management System


                1. Login
                2. Create Account
                3. Exit

   Enter your choice: _
```

1. **Account Creation**: Account is created by generating random numbers for the user, the user has to input his/her name and password.

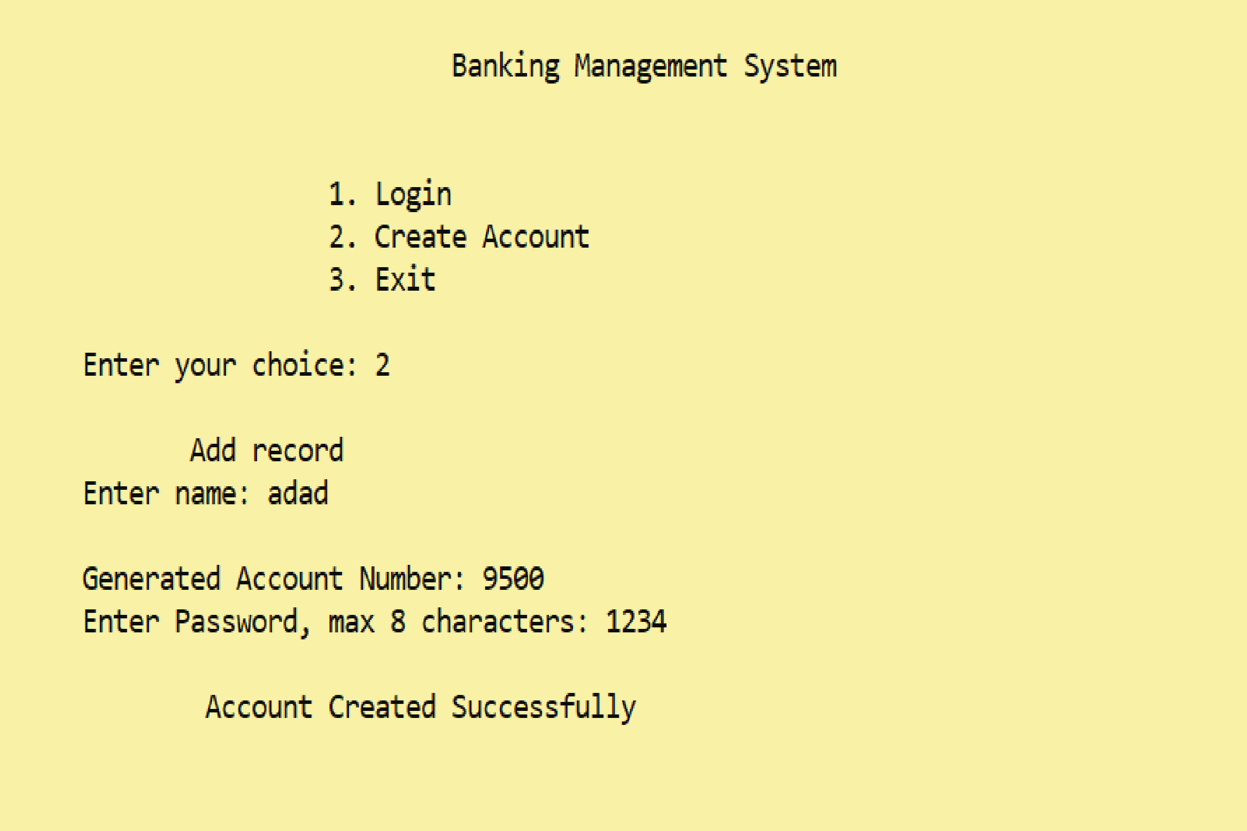
The function code:-

```
int generateRandomAccountNumber() {
    int accountNumber;
    do {
        accountNumber = (rand() % (MAX_ACCOUNT_NUMBER - MIN_ACCOUNT_NUMBER + 1)) + MIN_ACCOUNT_NUMBER;
    } while (accountExists(accountNumber));
    return accountNumber;
}
```

```
void createAccount() {
    FILE *info;
    struct user_account new_user;
    info = fopen("info.txt", "a");
    if (info == NULL) {
        printf("Error opening file!\n");
        return;
    }
    printf("\n\t       Add record      \n");
    printf("\tEnter name: ");
    getchar();
    scanf("%s", new_user.usr_name);
    new_user.id = generateRandomAccountNumber();
    printf("\n\tGenerated Account Number: %d\n", new_user.id);
    while (1) {
        printf("\tEnter Password, max 8 characters: ");
        scanf("%s", new_user.password);
        int len = strlen(new_user.password);
        if (len <= 8) {
            new_user.balance = 0;
            fprintf(info, "%s %d %s %.2f\n", new_user.usr_name,
new_user.id, new_user.password, new_user.balance);
```

```
            printf("\n\t\tAccount Created Successfully\n");
            break;
        } else {
            printf("\n\t\tPassword exceeded maximum length! Please set within 8 characters.\n");
        }
    }
    fclose(info);
    delay(2);
}
```

Input and Output:



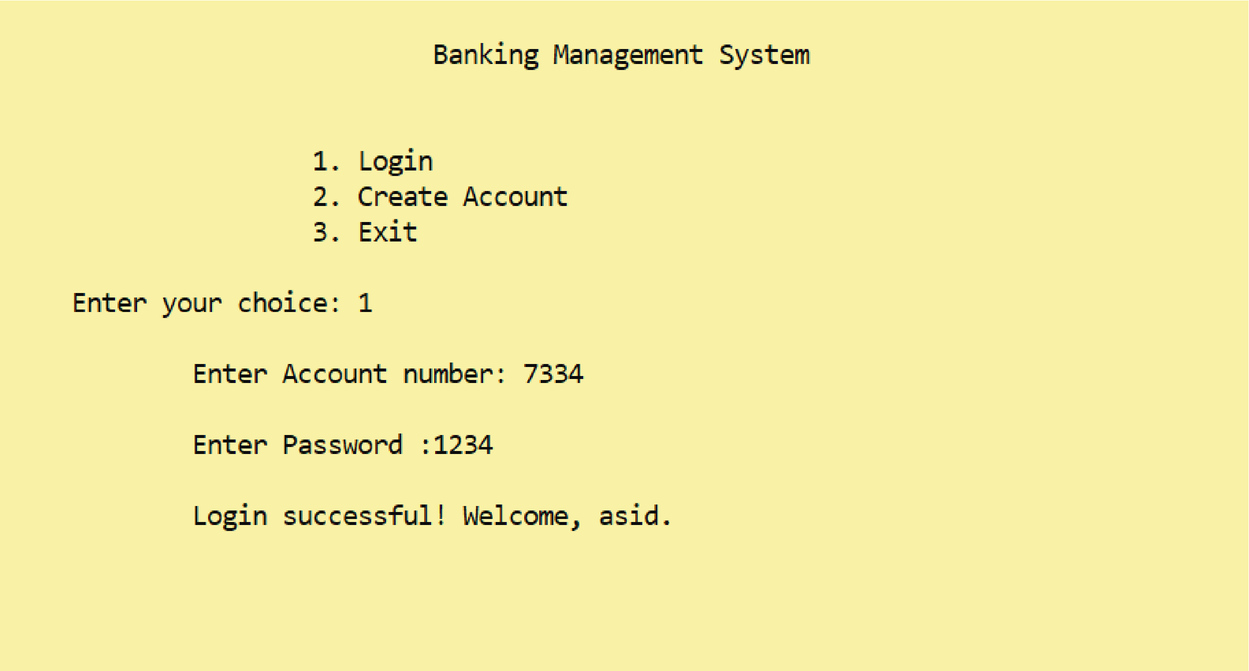
```
                          Banking Management System


                  1. Login
                  2. Create Account
                  3. Exit

     Enter your choice: 2

            Add record
     Enter name: adad

     Generated Account Number: 9500
     Enter Password, max 8 characters: 1234

             Account Created Successfully
```

2. **Login function**: After creating the account the user uses the given account number and password to enter the main functionality of the banking system.

The code:-

```
void user_menu(){


    int m;

    FILE *info;

    struct user_account user,check;


        system("cls");

        printf("     \n\t\t\t\tBanking Management System     \n\n");
        printf("\n\t\t\t1. Login");
        printf("\n\t\t\t2. Create Account\n");
        printf("\t\t\t3. Exit \n");

    printf("\n\tEnter your choice: ");
    scanf("%d", &m);

    switch (m) {
    case 1:
    usr_menu:
        info = fopen("info.txt","r");
        printf("\n\t\tEnter Account number: ");
        scanf("%d",&check.id);
        //getchar();
        fflush(stdin);
        printf("\n\t\tEnter Password :");
        scanf("%s",check.password);
        while(fscanf(info,"%s %d %s %f",user.usr_name,&user.id,&user.password,&user.balance)!= EOF){
            if( (check.id == user.id) && (strcmp(check.password, user.password) == 0)){
                printf("\n\t\tLogin successful! Welcome, %s.\n\n", user.usr_name);
                fclose(info);
                delay(2);
                login(user);
                //goto user_menu;
            }
        }
        printf("\n\t\tLogin Failed! Re-enter ID and Password.\n\n");
        fclose(info);
        delay(2);
        goto usr_menu;

        break;
```

Input and Output:-



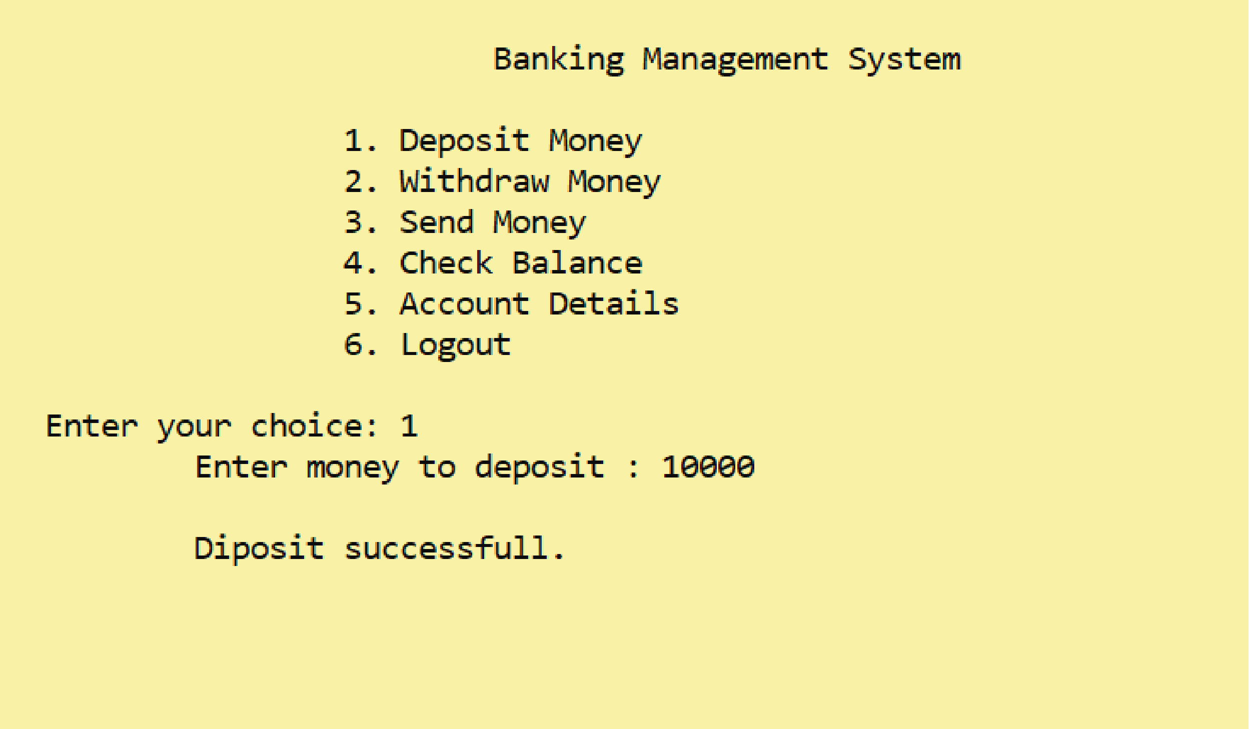
```
                           Banking Management System


                   1. Login
                   2. Create Account
                   3. Exit

  Enter your choice: 1

          Enter Account number: 7334

          Enter Password :1234

          Login successful! Welcome, asid.
```

3. **Deposit Functionality**: Allows users to deposit money into an account and updates the balance accordingly.

The code:-

```
void depositMoney(int id,float balance){
   char trans_type[20] = "DIPOSIT-";
   FILE *info;
   FILE *newinfo;
   FILE *trans;
   trans = fopen("transaction.txt","a");
   info = fopen("info.txt","r");
   newinfo = fopen("newinfo.txt","w");
            while(fscanf(info,"%s %d %s %f",user.usr_name,&user.id,&user.password,&user.balance)!= EOF){
               if(id == user.id){
                  user.balance += balance;
            fprintf(trans,"%s %d %s %.2f %s\n",user.usr_name,user.id,trans_type,balance,timeDate());
          }
          fprintf(newinfo,"%s %d %s %.2f\n",user.usr_name,user.id,user.password,user.balance);


        }
  fclose(trans);
  fclose(newinfo);
  fclose(info);
  remove("info.txt");
  rename("newinfo.txt", "info.txt");
}
```

Input and Output:-



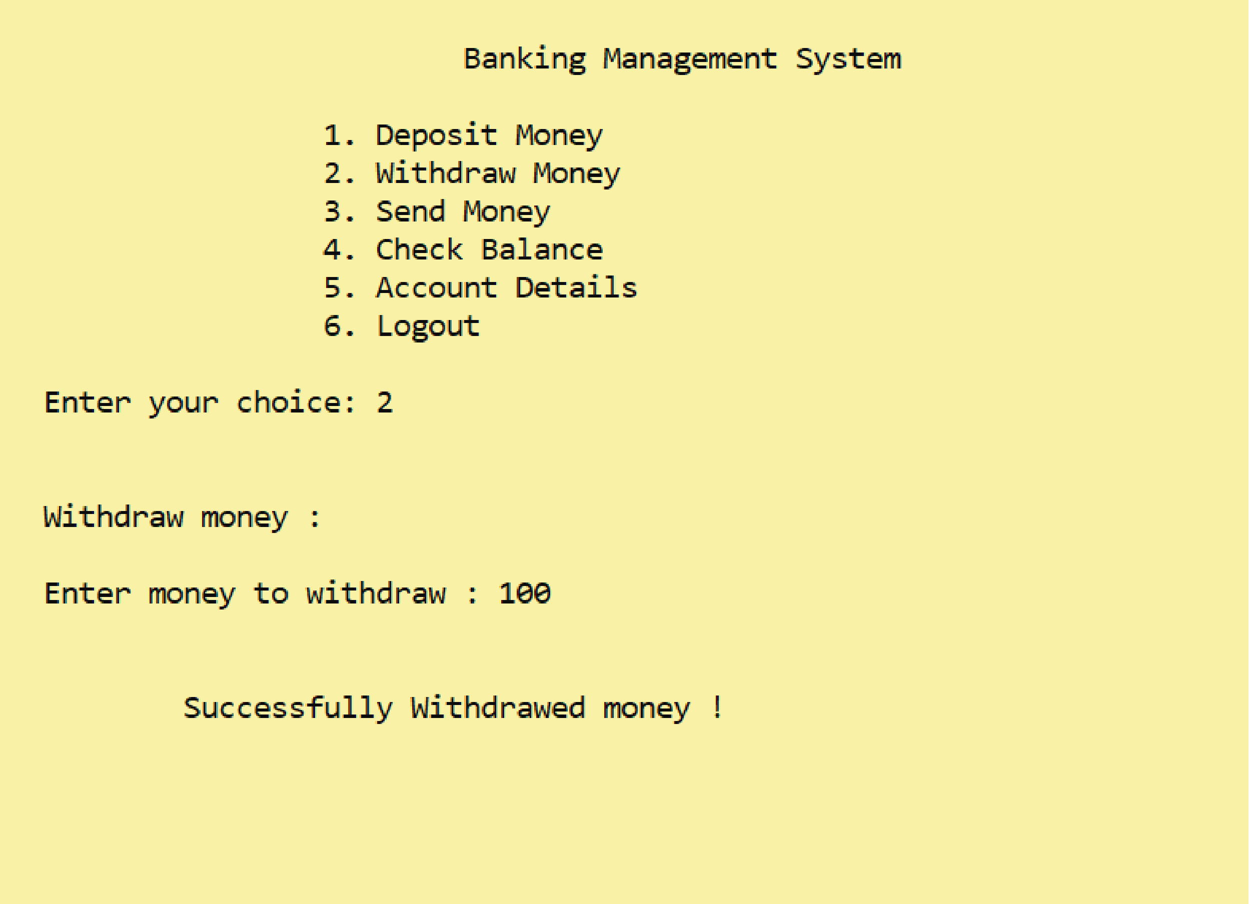
```
                          Banking Management System

                  1. Deposit Money
                  2. Withdraw Money
                  3. Send Money
                  4. Check Balance
                  5. Account Details
                  6. Logout

  Enter your choice: 1
          Enter money to deposit : 10000

          Diposit successfull.
```

4. **Withdrawal Functionality**: Users can withdraw money from an account with balance checks to prevent overdrafts.

The code:-

```
void withdrawMoney(int id,double balance){

    struct user_account user;

    char trans_type[20] = "WITHDRAW";

    FILE *info,*newinfo,*trans;

    trans = fopen("transaction.txt","a");

    info = fopen("info.txt","r");

    newinfo = fopen("newinfo.txt","w");


    while(fscanf(info,"%s %d %s %f",user.usr_name,&user.id,&user.password,&user.balance)!=
EOF){

        if(id == user.id){

            if(balance > user.balance){

                printf("\n\t\t\tInsufficient Balance!!\n\n");
            }else{
                user.balance -= balance;
                fprintf(trans,"%s %s %d %s %.2f \n",timeDate(),user.usr_name,user.id,trans_type,balance);
            }
        }
        fprintf(newinfo,"%s %d %s %.2f\n",user.usr_name,user.id,user.password,user.balance);
    }
    fclose(trans);
    fclose(newinfo);
    fclose(info);

    remove("info.txt");
    rename("newinfo.txt","info.txt");
}
```

Input and Output:-



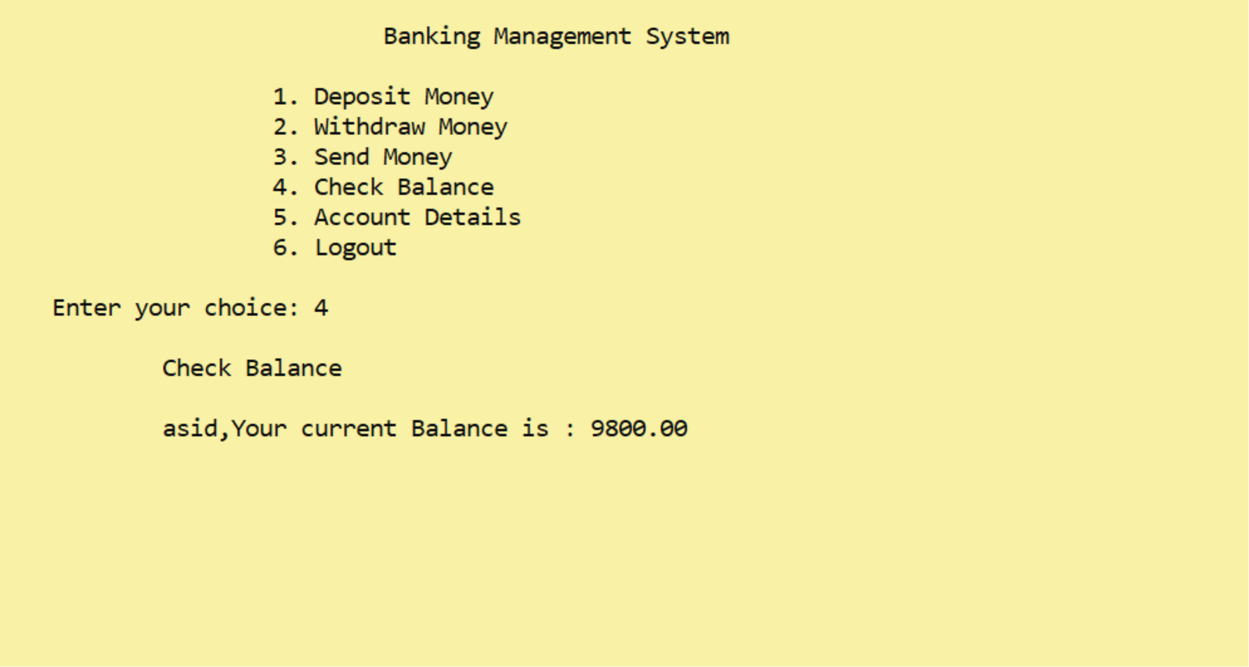
```
                        Banking Management System

                1. Deposit Money
                2. Withdraw Money
                3. Send Money
                4. Check Balance
                5. Account Details
                6. Logout

Enter your choice: 2


Withdraw money :

Enter money to withdraw : 100


        Successfully Withdrawed money !
```

5. **Balance Inquiry**: Displays the current balance of a specified account.

The code:-

```
void checkBalance(int id){


   struct user_account user;

   FILE *info;

   info = fopen("info.txt","r");

   printf("\n\t\tCheck Balance\t\n");


   while(fscanf(info,"%s %d %s %f",user.usr_name,&user.id,&user.password,&user.balance)!=
EOF){

       if(id == user.id){

           printf("\n\t\t%s,Your current Balance is : %.2f\n\n\n",user.usr_name,user.balance);

           break;

       }

   }


   fclose(info);

}
```

Input and Output:-



```
                        Banking Management System

                1. Deposit Money
                2. Withdraw Money
                3. Send Money
                4. Check Balance
                5. Account Details
                6. Logout

Enter your choice: 4

        Check Balance

        asid,Your current Balance is : 9800.00
```

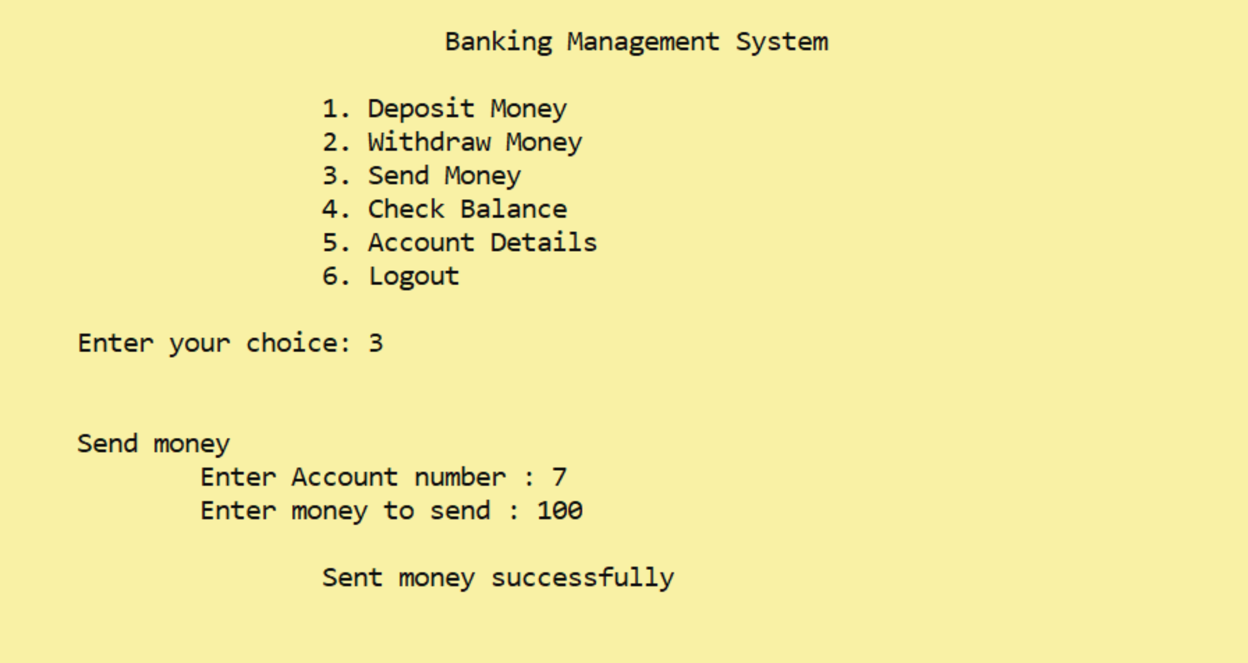
6. Transaction: Allows user to transact to another account. It enables the user to send money.

Function code:-

```
void sendMoney(int sen_id, int rec_id, float balance) {


   struct user_account user;
   FILE *info;
   info = fopen("info.txt", "r");
   while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, &user.password, &user.balance) != EOF) {
         if(sen_id == user.id){
            fclose(info);
            if(balance > user.balance){
               printf("\n\t\t\tInsufficient Balance!!\n\n");
            }else{
               if(accountExists(rec_id)){
                  withdrawMoney(sen_id,balance);
                  depositMoney(rec_id,balance);
                  printf("\n\t\t\tSent money successfully");


               }
               else{
                  printf("\n\t\tRecipient account not found.\n\n");
               }
            }
```

```
      }


  }


}
```

Input and Output:-



```
                        Banking Management System

                1. Deposit Money
                2. Withdraw Money
                3. Send Money
                4. Check Balance
                5. Account Details
                6. Logout

Enter your choice: 3


Send money
        Enter Account number : 7
        Enter money to send : 100

                Sent money successfully
```

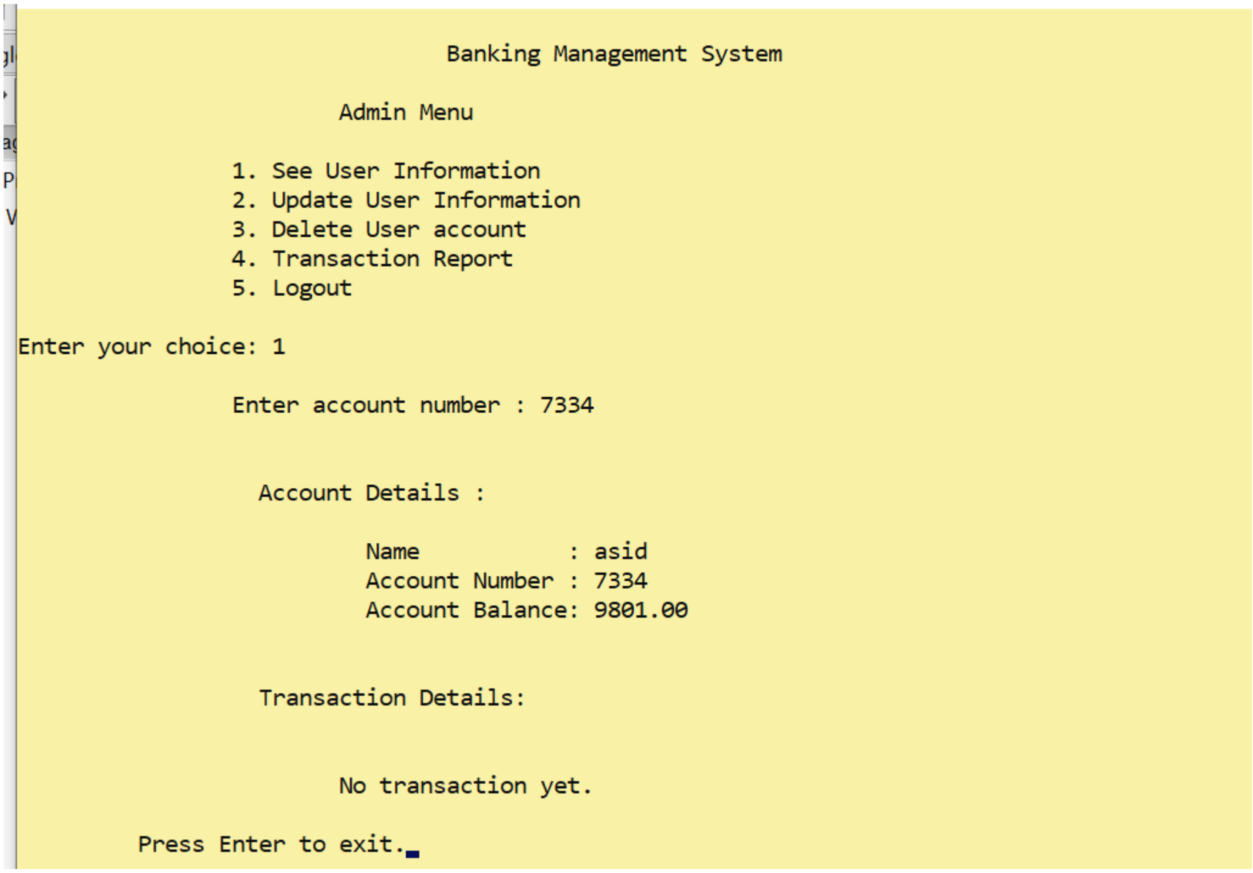
7. **Account details and Transaction History**: Stores a history of transactions, allowing users to track deposits and withdrawals for each account.

Function code:-

```
void accountDetails(int id) {
   struct user_account user;
   FILE *info, *trans;
   info = fopen("info.txt", "r");
   if(accountExists(id) == 0){
      printf("\n\n\t\tAccount Doesnot Exist....\n");
   }
   while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, &user.password,
&user.balance) != EOF) {
      if (id == user.id) {
         printf("\n\n\t\t  Account Details :\n\n\t\t\t  Name          : %s\n\t\t\t  Account
Number : %d\n\t\t\t  Account Balance: %.2f\n",user.usr_name,user.id,user.balance);
         break;
      }
   }
   fclose(info);

   char time[50], name[20];
   float balance;
```

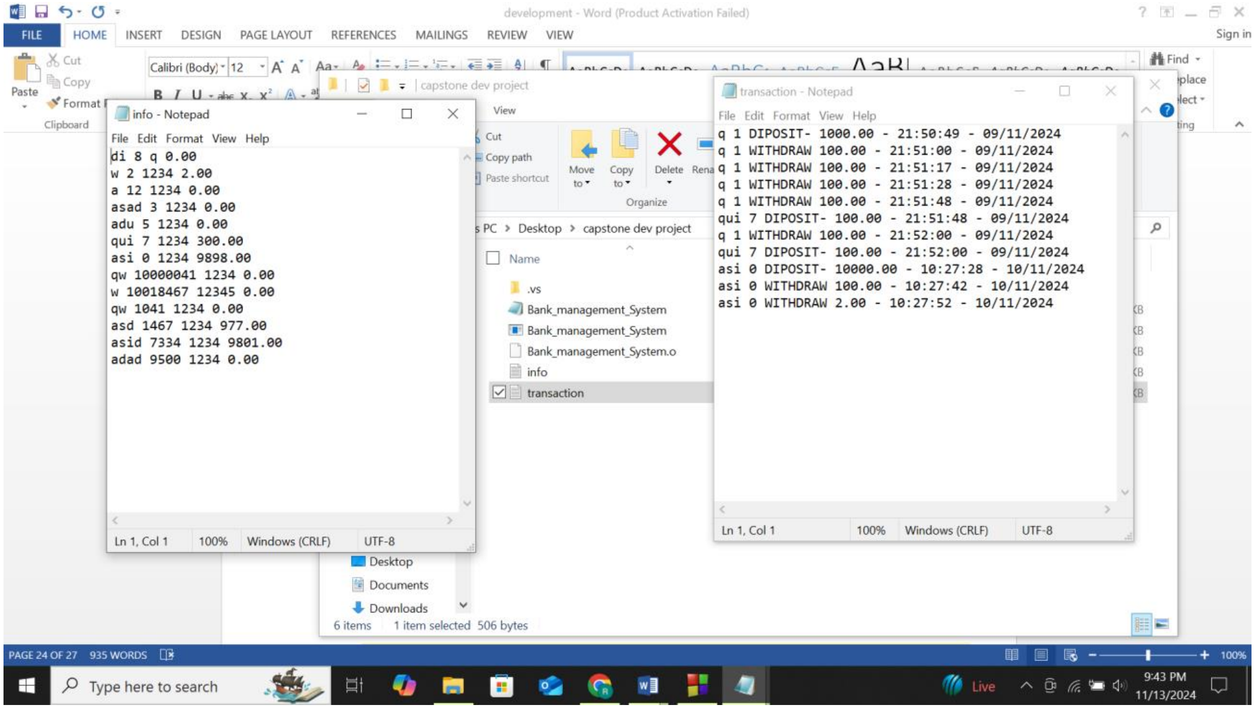
```
    int p_id, flag = 1;
    char trans_type[20];

    trans = fopen("transaction.txt", "r");
    printf("\n\n\t\t  Transaction Details:\n\n");
    while (fscanf(trans, "%s %d %s %f", name, &p_id, trans_type, &balance) != EOF) {
        fgets(time, sizeof(time), trans);
        time[strcspn(time, "\n")] = 0;
        if (id == p_id) {
            printf("\t\t%-10s %.2f at %s\n", trans_type, balance, time);
            flag = 0;
        }
    }
    if(flag){
        printf("\n\t\t\tNo transaction yet.\n");
    }
    fclose(trans);

    printf("\n\t Press Enter to exit.");
    getch();
    system("cls");
}
```

Input and output:-



```
                                Banking Management System

                        Admin Menu

                1. See User Information
                2. Update User Information
                3. Delete User account
                4. Transaction Report
                5. Logout

Enter your choice: 1

                Enter account number : 7334


                  Account Details :

                          Name           : asid
                          Account Number : 7334
                          Account Balance: 9801.00


                  Transaction Details:


                        No transaction yet.

         Press Enter to exit._
```

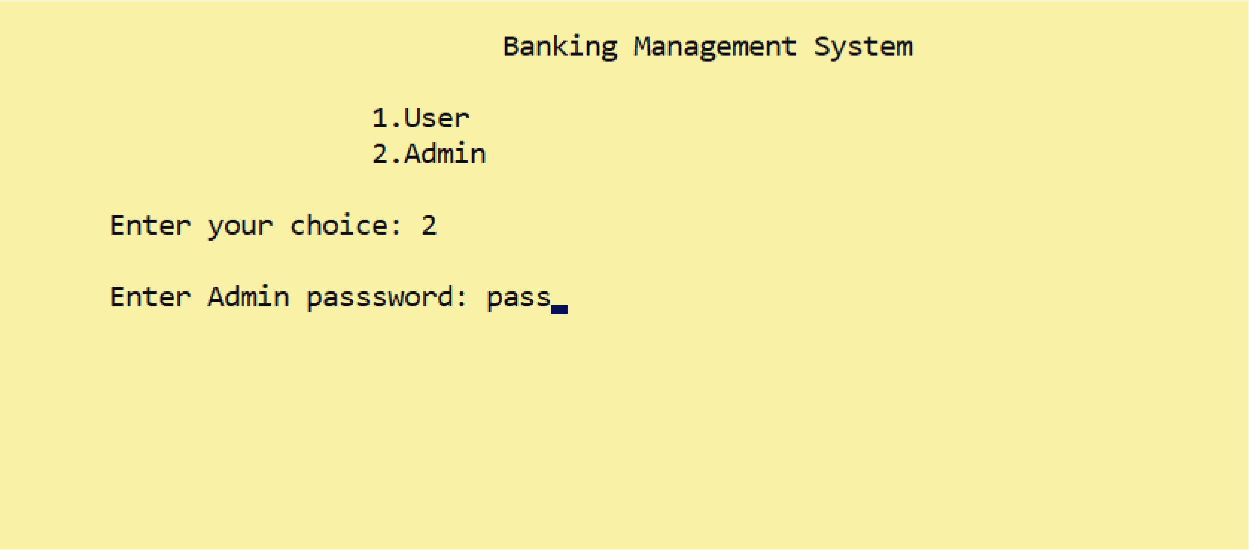
8. **File-Based Data Management**: Account data will be stored in text files, ensuring data persistence even after the program is closed.



```
di 8 q 0.00
w 2 1234 2.00
a 12 1234 0.00
asad 3 1234 0.00
adu 5 1234 0.00
qui 7 1234 300.00
asi 0 1234 9898.00
qw 10000041 1234 0.00
w 10018467 12345 0.00
qw 1041 1234 0.00
asd 1467 1234 977.00
asid 7334 1234 9801.00
adad 9500 1234 0.00
```

```
q 1 DIPOSIT- 1000.00 - 21:50:49 - 09/11/2024
q 1 WITHDRAW 100.00 - 21:51:00 - 09/11/2024
q 1 WITHDRAW 100.00 - 21:51:17 - 09/11/2024
q 1 WITHDRAW 100.00 - 21:51:28 - 09/11/2024
q 1 WITHDRAW 100.00 - 21:51:48 - 09/11/2024
qui 7 DIPOSIT- 100.00 - 21:51:48 - 09/11/2024
q 1 WITHDRAW 100.00 - 21:52:00 - 09/11/2024
qui 7 DIPOSIT- 100.00 - 21:52:00 - 09/11/2024
asi 0 DIPOSIT- 10000.00 - 10:27:28 - 10/11/2024
asi 0 WITHDRAW 100.00 - 10:27:42 - 10/11/2024
asi 0 WITHDRAW 2.00 - 10:27:52 - 10/11/2024
```

Admin----



```
                                Banking Management System

                        1.User
                        2.Admin

       Enter your choice: 2

       Enter Admin passsword: pass_
```

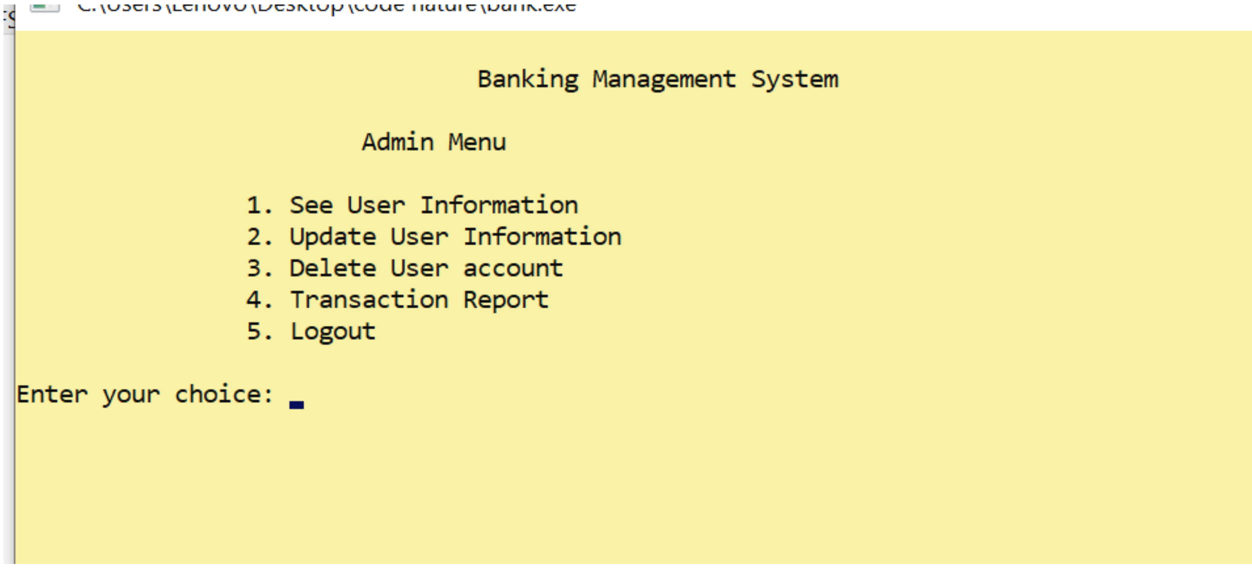
1. Admin menu—

Function Code:-

```
void adminMenu(){

  int m,id;
  while(1){

    printf("     \n\t\t\t\tBanking Management System     \n\n");
    printf("\t\t\tAdmin Menu \n\n");
    printf("\t\t1. See User Information\n");
    printf("\t\t2. Update User Information\n");
    printf("\t\t3. Delete User account\n");
    printf("\t\t4. Transaction Report \n");
    printf("\t\t5. Logout \n");
    printf("\nEnter your choice: ");
    scanf("%d", &m);
    switch(m){
      case 1:

        printf("\n\t\tEnter account number : ");
        scanf("%d",&id);
        if(accountExists(id)){
          accountDetails(id);
        }
```

```
                else{
                    printf("\n\t\tAccount doesn't Exist.Press Enter to proceed.");
                }
                getchar();
                system("cls");
                break;
            case 2:
                updateInfo();
                delay(2);
                system("cls");
                break;
            case 3:
                deleteAccount();
                system("cls");
                break;

            case 4:
                report();


                system("cls");
                break;

            case 5:
                printf("\n\t\tExiting ...\n\n");
                delay(2);
                system("cls");
                main();
                break;
            default:
                printf("\t\tInvalid choice! Please try again.\n");
                delay(2);
                system("cls");
                adminMenu();
                break;
        }
        system("cls");
    }
}
```

Code output:-



```
C:\Users\Lenovo\Desktop\code nature\bank.exe

                               Banking Management System

                       Admin Menu

               1. See User Information
               2. Update User Information
               3. Delete User account
               4. Transaction Report
               5. Logout

Enter your choice: _
```

2. User Information Checking

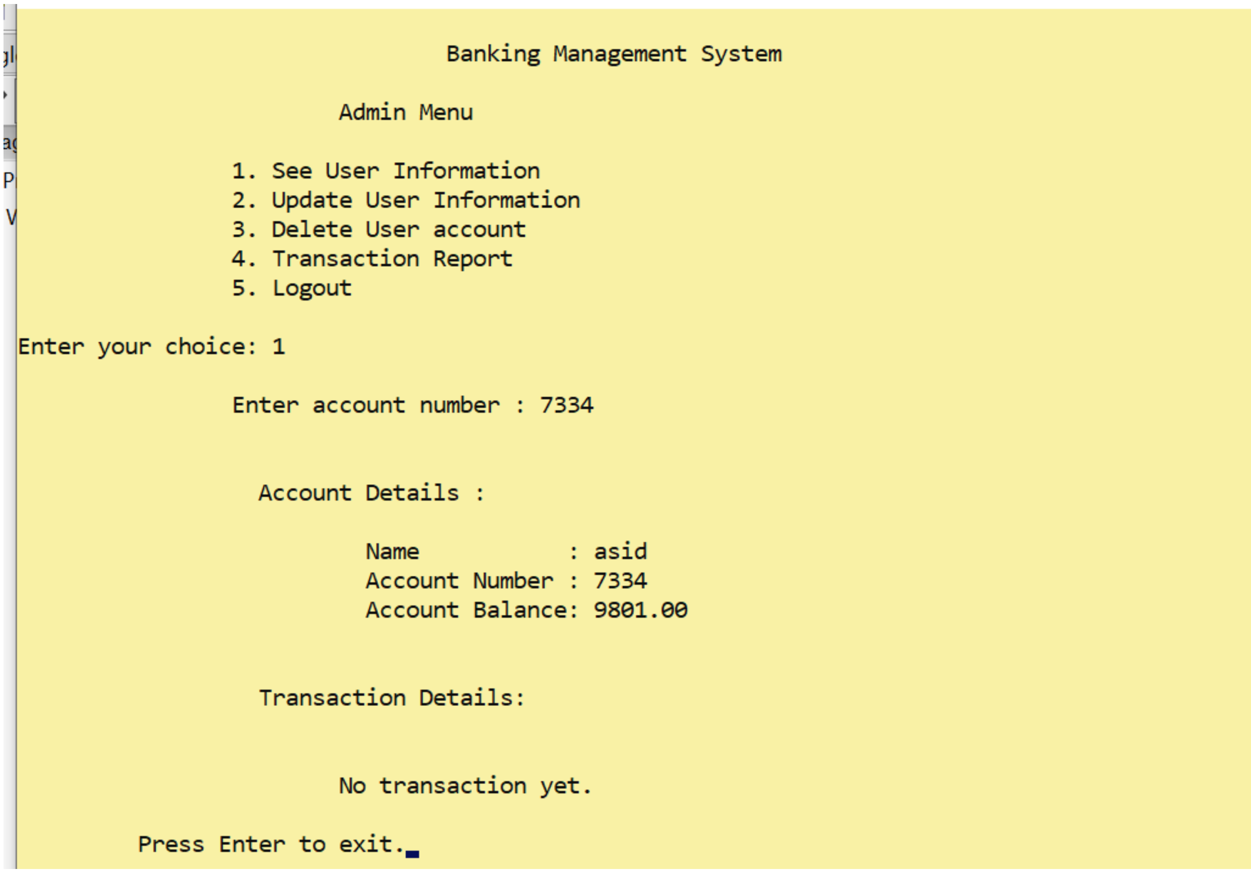
Functionality: Admin can check the user information by their account number.

Function code:-

```
void accountDetails(int id) {
    struct user_account user;
    FILE *info, *trans;
    info = fopen("info.txt", "r");
    if(accountExists(id) == 0){
        printf("\n\n\t\tAccount Doesnot Exist....\n");
    }
    while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, &user.password,
&user.balance) != EOF) {
        if (id == user.id) {
            printf("\n\n\t\t  Account Details :\n\n\t\t\t  Name           : %s\n\t\t\t
Account Number : %d\n\t\t\t  Account Balance:
%.2f\n",user.usr_name,user.id,user.balance);
            break;
        }
    }
    fclose(info);

    char time[50], name[20];
    float balance;
    int p_id, flag = 1;
    char trans_type[20];

    trans = fopen("transaction.txt", "r");
    printf("\n\n\t\t  Transaction Details:\n\n");
    while (fscanf(trans, "%s %d %s %f", name, &p_id, trans_type, &balance) !=
EOF) {
        fgets(time, sizeof(time), trans);
        time[strcspn(time, "\n")] = 0;
        if (id == p_id) {
            printf("\t\t%-10s %.2f at %s\n", trans_type, balance, time);
            flag = 0;
        }
    }
    if(flag){
        printf("\n\t\t\tNo transaction yet.\n");
    }
    fclose(trans);

    printf("\n\t Press Enter to exit.");
    getch();
```

```
      system("cls");
    }
```

Input and Output:-



```
                                 Banking Management System

                         Admin Menu

                 1. See User Information
                 2. Update User Information
                 3. Delete User account
                 4. Transaction Report
                 5. Logout

Enter your choice: 1

                 Enter account number : 7334


                   Account Details :

                           Name           : asid
                           Account Number : 7334
                           Account Balance: 9801.00


                   Transaction Details:


                         No transaction yet.

          Press Enter to exit.
```

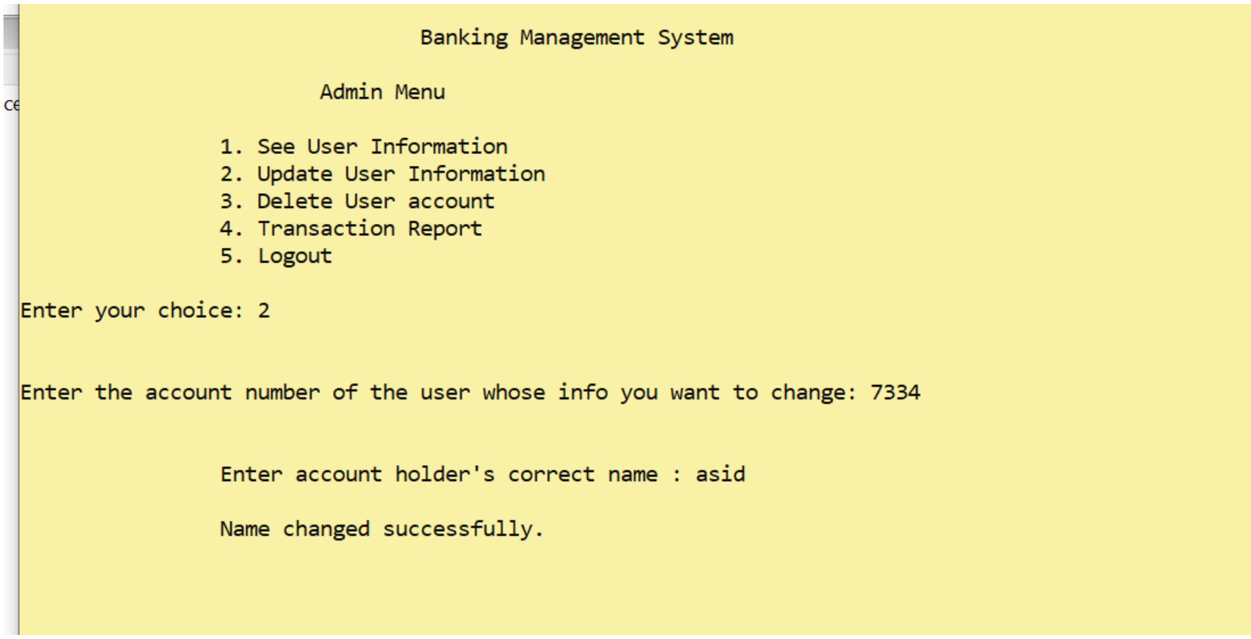
3. Update User Information: Update the user name from the account.

Function code:-

```
void updateInfo() {
   struct user_account user, check;
   FILE *info, *newinfo;
   int flag = 1;
   info = fopen("info.txt", "r");
   newinfo = fopen("newinfo.txt", "w");

   printf("\n\nEnter the account number of the user whose info you want to change: ");
   scanf("%d", &check.id);
```

```
    while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, user.password,
&user.balance) != EOF) {
        if (check.id == user.id) {
            flag = 0;
            printf("\n\n\t\tEnter account holder's correct name : ");

            getchar();
            fgets(check.usr_name, sizeof(check.usr_name), stdin);
            if (check.usr_name[strlen(check.usr_name) - 1] == '\n') {
                check.usr_name[strlen(check.usr_name) - 1] = '\0';
            }
            strcpy(user.usr_name, check.usr_name);

            printf("\n\t\tName changed successfully.\n\n");
        }
        fprintf(newinfo, "%s %d %s %.2f\n", user.usr_name, user.id, user.password,
user.balance);

    }
    if(flag){
        printf("\n\t\tAccount doesn't exists....\n\n");
        delay(2);
    }

    fclose(newinfo);
    fclose(info);

    remove("info.txt");
    rename("newinfo.txt", "info.txt");

}
```

Input Output:-



```
                                Banking Management System

                        Admin Menu

                1. See User Information
                2. Update User Information
                3. Delete User account
                4. Transaction Report
                5. Logout

Enter your choice: 2


Enter the account number of the user whose info you want to change: 7334


                Enter account holder's correct name : asid

                Name changed successfully.
```

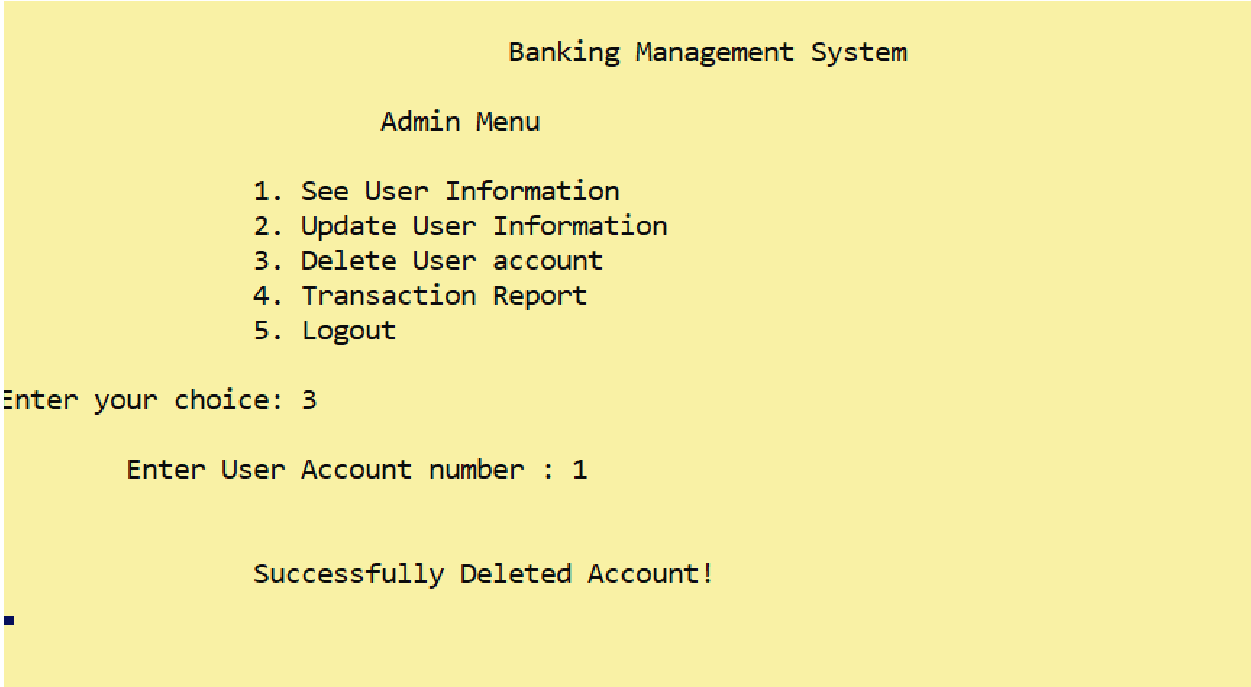
4. **Delete User account:** Admin can also delete the account of a user.

Code:-

```
void deleteAccount() {
  struct user_account user;
  int id;
  FILE *info, *newinfo;
  info = fopen("info.txt", "r");
  newinfo = fopen("newinfo.txt", "w");

  printf("\n\tEnter User Account number : ");
  scanf("%d", &id);

  int accountFound = 1;

  while (fscanf(info, "%s %d %s %f", user.usr_name, &user.id, user.password,
&user.balance) != EOF) {
      if (id == user.id) {
         accountFound = 0;
         continue;
      }
      fprintf(newinfo, "%s %d %s %.2f\n", user.usr_name, user.id,
user.password, user.balance);
```

```
   }

   if(accountFound){
      printf("\n\n\t\tAccount doesn't Exist.\n");
   }
   else{
      printf("\n\n\t\tSuccessfully Deleted Account!\n");
   }



   fclose(newinfo);
   fclose(info);

   //if (accountFound) {
      remove("info.txt");
      rename("newinfo.txt", "info.txt");

  // } else {
      //printf("\n\t\t Account no. %d is not found.\n", id);
      remove("newinfo.txt");
   //}
   delay(2);

}
```

Input and Output:-



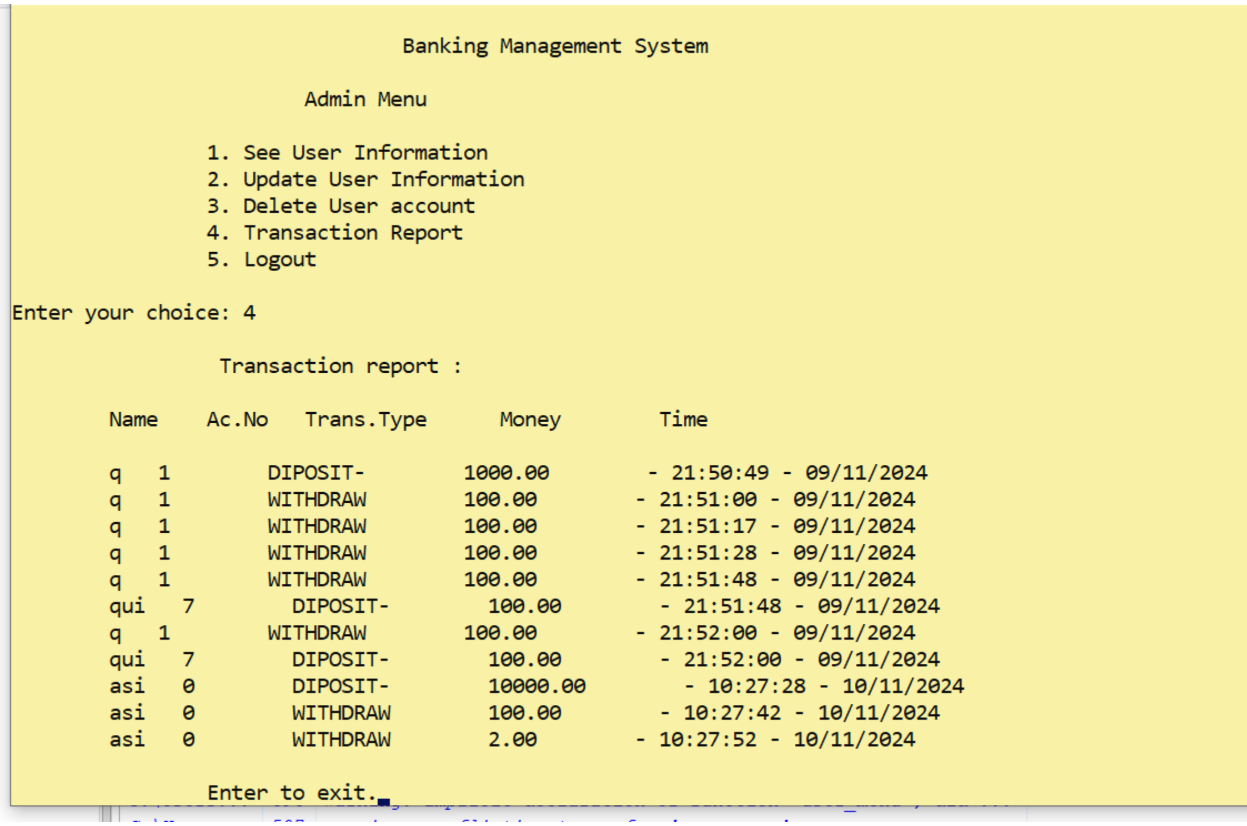
```
                               Banking Management System

                       Admin Menu

               1. See User Information
               2. Update User Information
               3. Delete User account
               4. Transaction Report
               5. Logout

Enter your choice: 3

       Enter User Account number : 1


               Successfully Deleted Account!
```

5. **Transaction Report:** Admin can view the transaction report of various users.

Function code:-

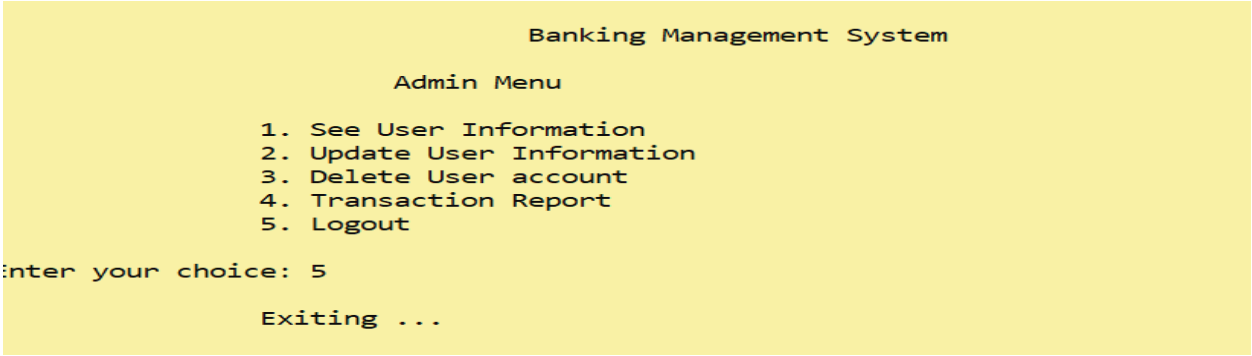
```
void report(){
   FILE *trans;
   char trans_type[20],time[70],name[20];
   float balance;
   int id;
   trans = fopen("transaction.txt","r");
   printf("\n\t\t Transaction report :\n");
   printf("\n\tName\tAc.No \tTrans.Type   \tMoney       Time\n\n");
   while(fscanf(trans,"%s %d %s %f",name,&id ,trans_type, &balance) != EOF){
      fgets(time, sizeof(time), trans);
      time[strcspn(time, "\n")] = 0;
      printf("\t%s   %d       %s       %.2f      %s\n",name,id,trans_type, balance,
time);
   }
   fclose(trans);
   printf("\n\t\tEnter to exit.");
   getch();

}
```

Input and output:-



```
                                Banking Management System

                        Admin Menu

                1. See User Information
                2. Update User Information
                3. Delete User account
                4. Transaction Report
                5. Logout

Enter your choice: 4

                 Transaction report :

        Name    Ac.No   Trans.Type      Money        Time

        q   1        DIPOSIT-       1000.00        - 21:50:49 - 09/11/2024
        q   1        WITHDRAW       100.00        - 21:51:00 - 09/11/2024
        q   1        WITHDRAW       100.00        - 21:51:17 - 09/11/2024
        q   1        WITHDRAW       100.00        - 21:51:28 - 09/11/2024
        q   1        WITHDRAW       100.00        - 21:51:48 - 09/11/2024
        qui   7        DIPOSIT-       100.00        - 21:51:48 - 09/11/2024
        q   1        WITHDRAW       100.00        - 21:52:00 - 09/11/2024
        qui   7        DIPOSIT-       100.00        - 21:52:00 - 09/11/2024
        asi   0        DIPOSIT-       10000.00        - 10:27:28 - 10/11/2024
        asi   0        WITHDRAW       100.00        - 10:27:42 - 10/11/2024
        asi   0        WITHDRAW       2.00        - 10:27:52 - 10/11/2024

                Enter to exit.
```

6. Logout



```
                                Banking Management System

                        Admin Menu

                1. See User Information
                2. Update User Information
                3. Delete User account
                4. Transaction Report
                5. Logout

nter your choice: 5

                Exiting ...
```