

MEDI - CAPS UNIVERSITY

INDORE



DEPARTMENT OF COMPUTER SCIENCE
BRANCH : CS - DATA SCIENCE

PROJECT FILE
ON
CARDLESS ATM TRANSACTION SYSTEM

SUBMITTED TO :

Ms. UPASANA TIWARI

SUBMITTED BY :

RITIK RATNAWAT

UTKARSH YEOLE

VEDANT DESHMUKH

YASH AGRAWAL

ACKNOWLEDGEMENT

We have successfully completed our project.

However, it would not have been possible without the support and help of many and the organization we would like to extend my sincere thanks to all of them.

We are highly indebted to **Mrs. Upasana Tiwari** (Assistant Professor of CSE Department, Medi - Caps University, Indore) for her guidance and constant supervision as well as for providing necessary information regarding the project and also for support in completing the project.

We wish to express our deep sense of gratitude to **Mr. D. K. Panda**, Dean of Engineering, for his guidance and useful suggestions which helped us in completing the project work on time.

Our heartiest thanks to the college management for providing technical facility.

CERTIFICATE

This to certify that the report of the project submitted is the outcome of the project work entitled **Cardless ATM Transaction System** carried out by **Ritik Ratnawat, Utkarsh Yeole, Vedant Deshmukh and Yash Agrawal** bearing Enrollment No. **EN19CS303041, EN19CS303055, EN19CS303056, EN19CS303060** respectively under my guidance and supervision for the award of Degree in Bachelor of Technology in Computer Science (Specialization in Data Science) of Medi – Caps University, Indore, Madhya Pradesh.

To the best of the my knowledge the report

- i) Embodies the work of the candidates.
- ii) Has duly been completed,
- iii) Fulfils the requirement of the ordinance relating to the B.Tech degree of the University and
- iv) Is up to the desired standard for the purpose of which is submitted.

(Signature of the Professor)

Mrs. Upasana Tiwari

Asst. Professor

Department of CSE

CONTENTS

➤ Abstract	-----	1
➤ Introduction	-----	2
➤ Project Description	-----	3
➤ Scope	-----	3
➤ Objective	-----	4
➤ Tools, Platform & Language	-----	4
➤ Technical Feasibility	-----	5
➤ Social Feasibility	-----	5
➤ Economic Feasibility	-----	5
➤ Operation Feasibility	-----	5
➤ Structure Chart	-----	6
➤ Modules in Detail	-----	7 - 20
✓ Home Screen	-----	7
✓ Login Page	-----	8 - 9
✓ Transaction Options Screen	-----	10 - 11
✓ Deposit Screen	-----	12 - 13
✓ Withdraw Screen	-----	14 - 15
✓ Check Balance Screen	-----	16 - 17
✓ Account Details Screen	-----	18 - 19
✓ Thank You Screen	-----	20
➤ Source Code	-----	21 - 30
➤ Advantages	-----	31
➤ Conclusion	-----	32
➤ References	-----	32

ABSTRACT

' CARDLESS ATM TRANSACTION SYSTEM '

The Cardless ATM Transaction System is the project which helps people to access their bank accounts in order to make cash withdrawals and transactions. Whenever the user need to make cash transactions, they can enter their PIN number (personal identification number) , make their choices, perform desired actions and then exit the software.

The ATM will service one customer at a time. A customer will be required to enter ATM personal identification number (PIN). The customer will then be able to perform one or more transactions. In order to complete the transaction, he need to enter a second pin called Transaction Pin. Customer will also be able to keep a check on his bank account through the ATM.



INTRODUCTION

As the time passes, human life is becoming more fast paced , stressful , and panicky. Humans have to constantly be aware, update themselves and adapt to the emerging requirements of this constantly modernising world. This becomes very difficult for a human to keep working, managing a family, and keep learning and updating himself. People tend to forget important things and tasks due to their stressful lives.



As we are growing as technical world, use of ATM cards and other payment services are grown in very short time everywhere but in the hustle and bustle of the city life we often tend to forget our wallets or card holder or sometimes we loose it. Also sometimes the other payment services are available or under maintenance which creates a terrible situation for the User.

Here we come up with an idea aims to solve this problem by adding up an another option for the Users for making a transaction (either deposit or withdraw). This project is also concerned about user account security and to help in an emergency situation.

PROJECT DESCRIPTION

NEED FOR THE SOFTWARE :

Now-a-days, everyone is very busy in their work. So they feel that the job must be easier so the system is used to reduce their work which is done in the ATM system. Instead of keeping lots of paper into a record or file, (it may be missed somewhere) this system help to keep the record of the customer it also keeps the details of the customer. It is also easy to access. Also in the emergency situation, User can have another option to deposit or withdraw cash.

PROBLEM DESCRIPTION :

The system mainly used by the bank clients. It reduces the time consumption and lot of paperwork. For any single operation it involves numerous references and updating also takes subsequent changes in other places.

SCOPE

There is a scope of following points in our project -

- It can be implemented in ATM machine by owner of bank or in charge of branch.
- It also adds up a second level of security.
- It is easy to learn the task for any programmer.
- It is a better facility in emergency situation.

OBJECTIVE

Our main objective is to speed up the transactions done by customers without the requirement of credit card by the user with an additional level of security. The second objective is to save the time which is very important now-a-days. This method will also help users that had forgot their ATM cards somewhere.

TOOLS, PLATFORM & LANGUAGE

TOOLS :

- ✓ Processor :- Intel Core i3 or Later or Compatible
- ✓ Hard Disk :- 10GB or more
- ✓ RAM :- 256MB or more
- ✓ Printer :- Any
- ✓ Monitor :- SVGA Color Monitor (Touch Screen or Simple)

PLATFORM:

- ✓ Operating System :- Microsoft Windows 7 or Later or Equivalent

LANGUAGES:

- ✓ Front End :- C/C++ Graphics
- ✓ Back End :- C Programming

TECHNICAL FEASIBILITY

The system is being developed in C language. The compiler being used for the creation of the program is Turbo C++ (version 3.2). It provides comprehensive function to make it user friendly. The data entry and report generation is also made easy. The machine configuration also supports this software. It is easily capable of handling the efficient execution of this program.

SOCIAL FEASIBILITY

As this system is user friendly and flexible, some problems will also be solved which employee may be facing when using existing system. So we can say that system is socially feasible.

ECONOMIC FEASIBILITY

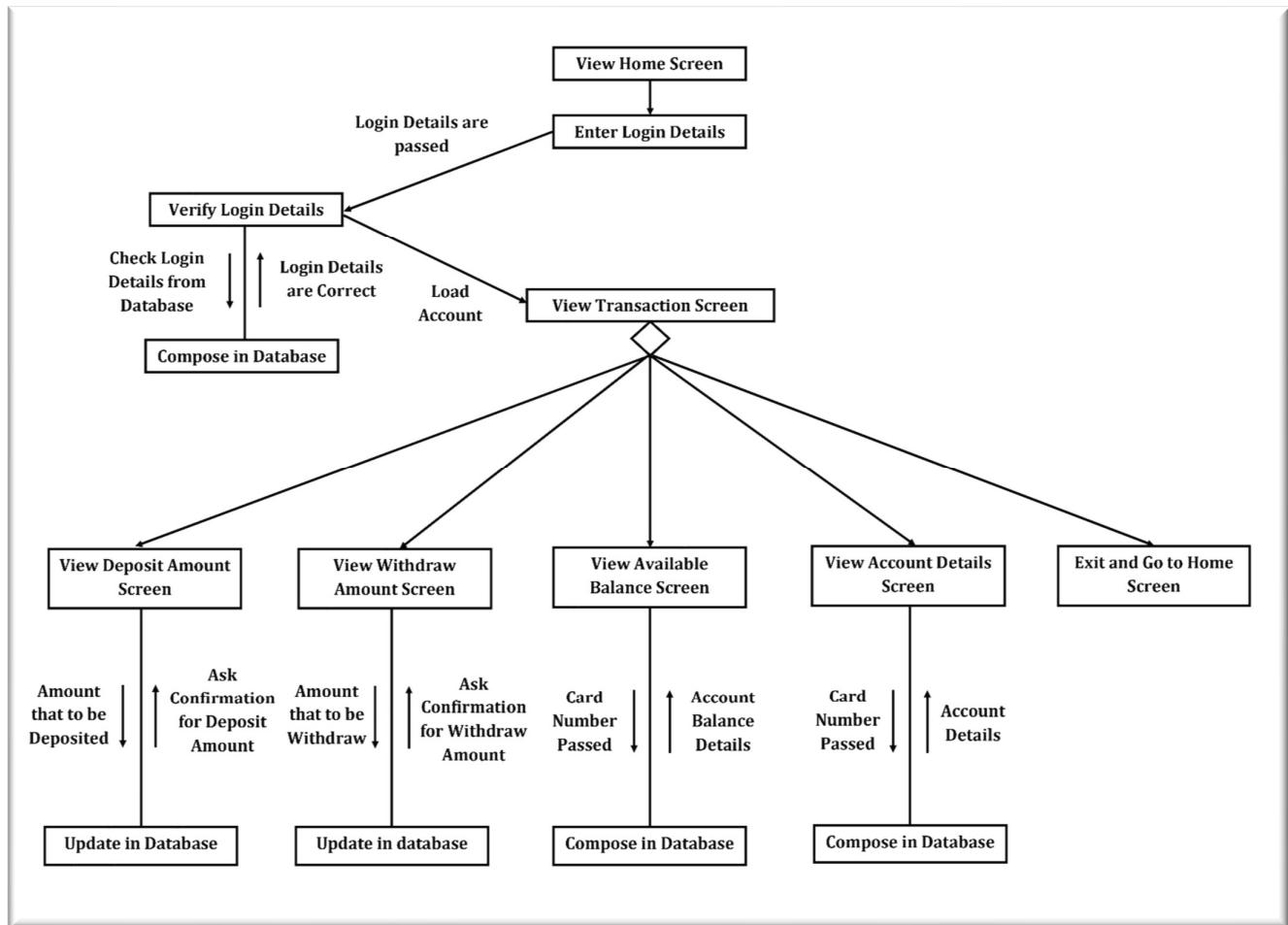
The cost of converting from manual system to new automatic computerized system is not probably more. For construction of the new system, the rooms and its facilities are available so it does not require any extra resource, only the software requirement is there. Also, the compiler used here, Turbo C++, is completely free to download and use.

OPERATION FEASIBILITY

Since the system is being in user friendly way, the new customers within a few time can master it.

STRUCTURE CHART

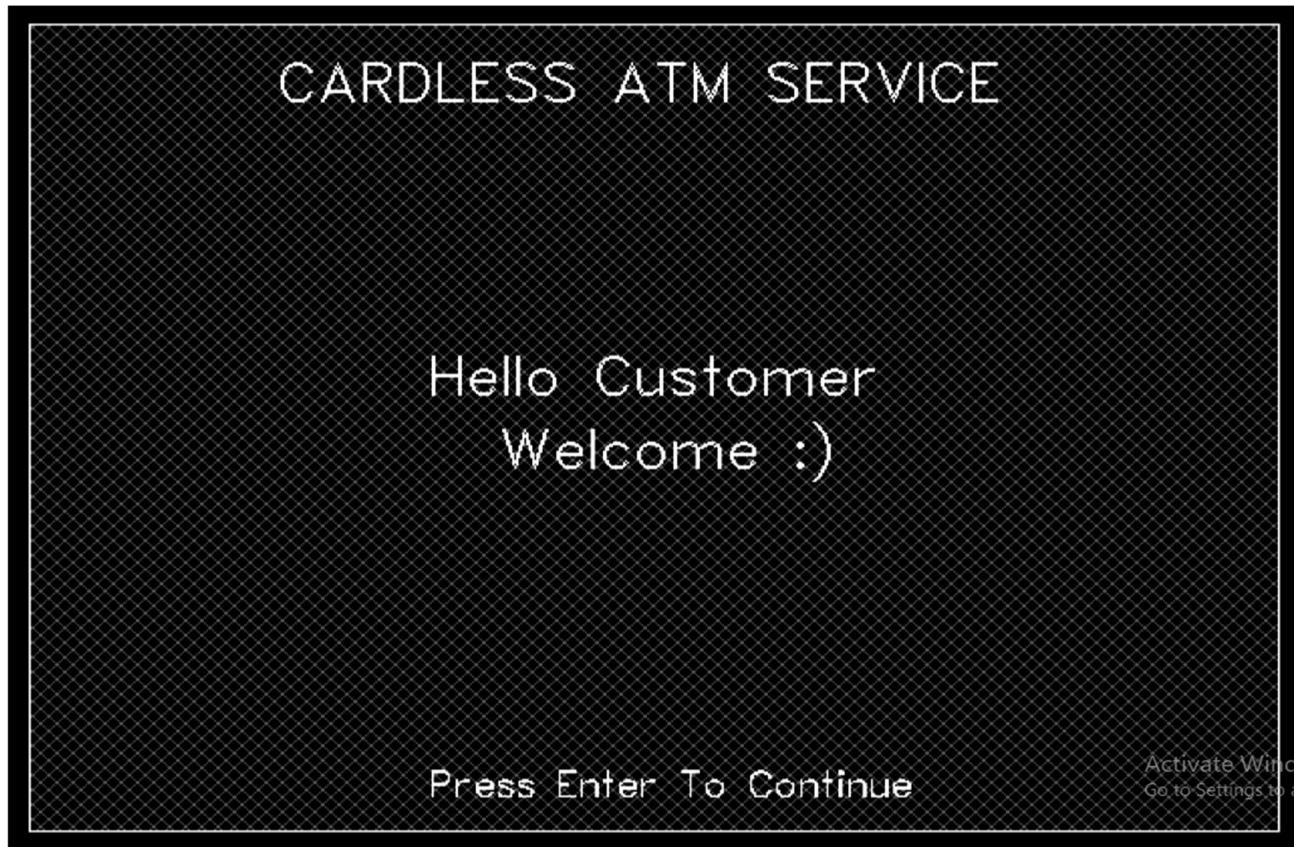
The following flowchart gives us the complete overview of the complete working of the program. Basic functionality of each module is represented on the diagram below. We will study how each of the individual modules functions in detail later.



MODULES IN DETAIL

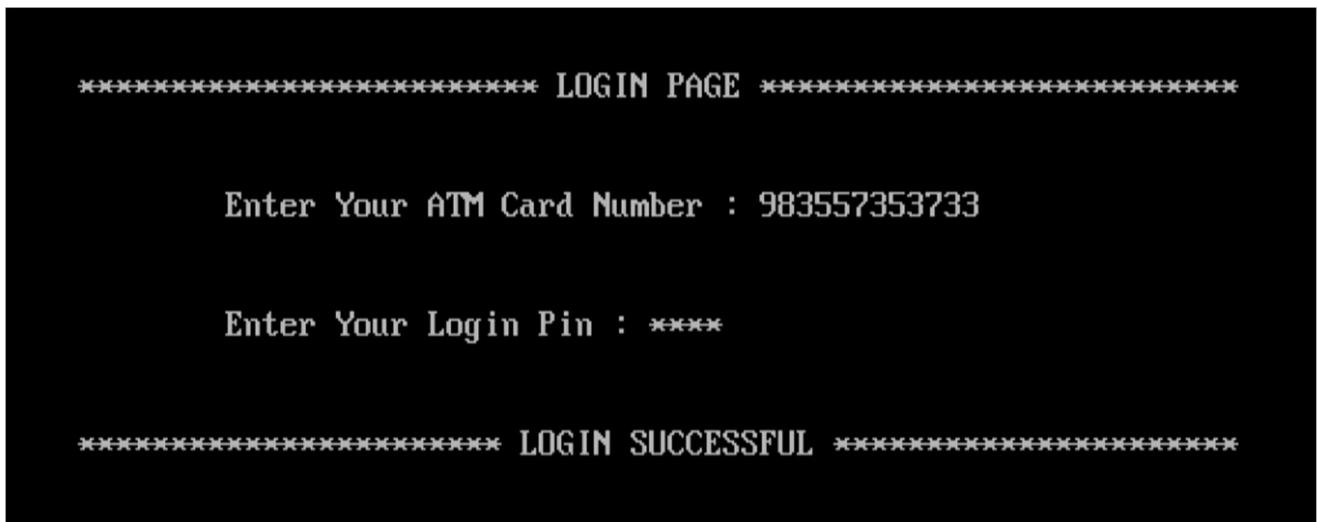
HOME SCREEN →

This is the first screen user sees. This screen is called the '**Home screen**'. We have created this screen using graphics in C language. The user press ENTER to go to the next screen.

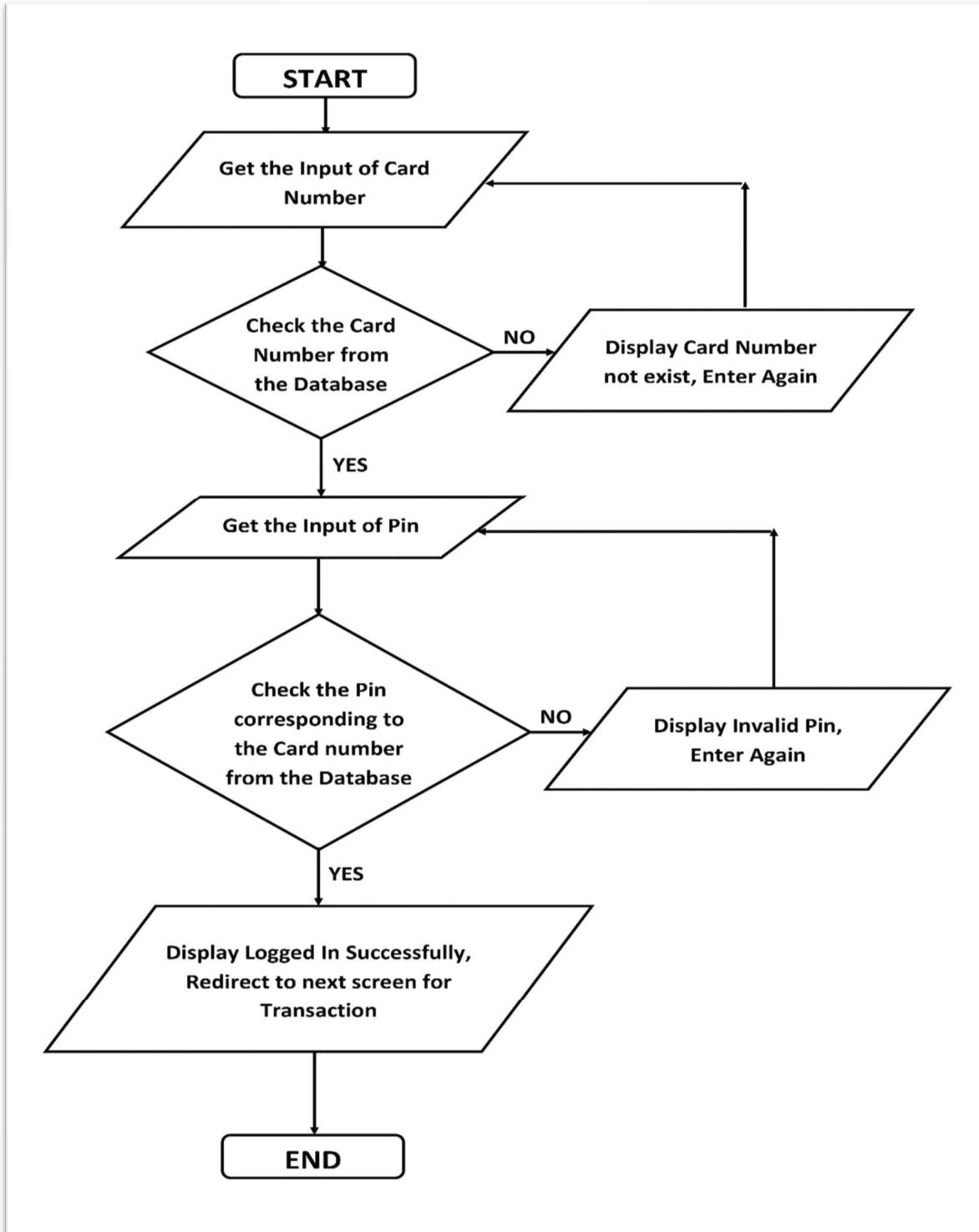


LOGIN PAGE →

This is the second screen. Here, the user needs to enter his correct ATM card number and Login Pin to access his account and proceeds towards performing transactions. The flowchart explains the process in detail.

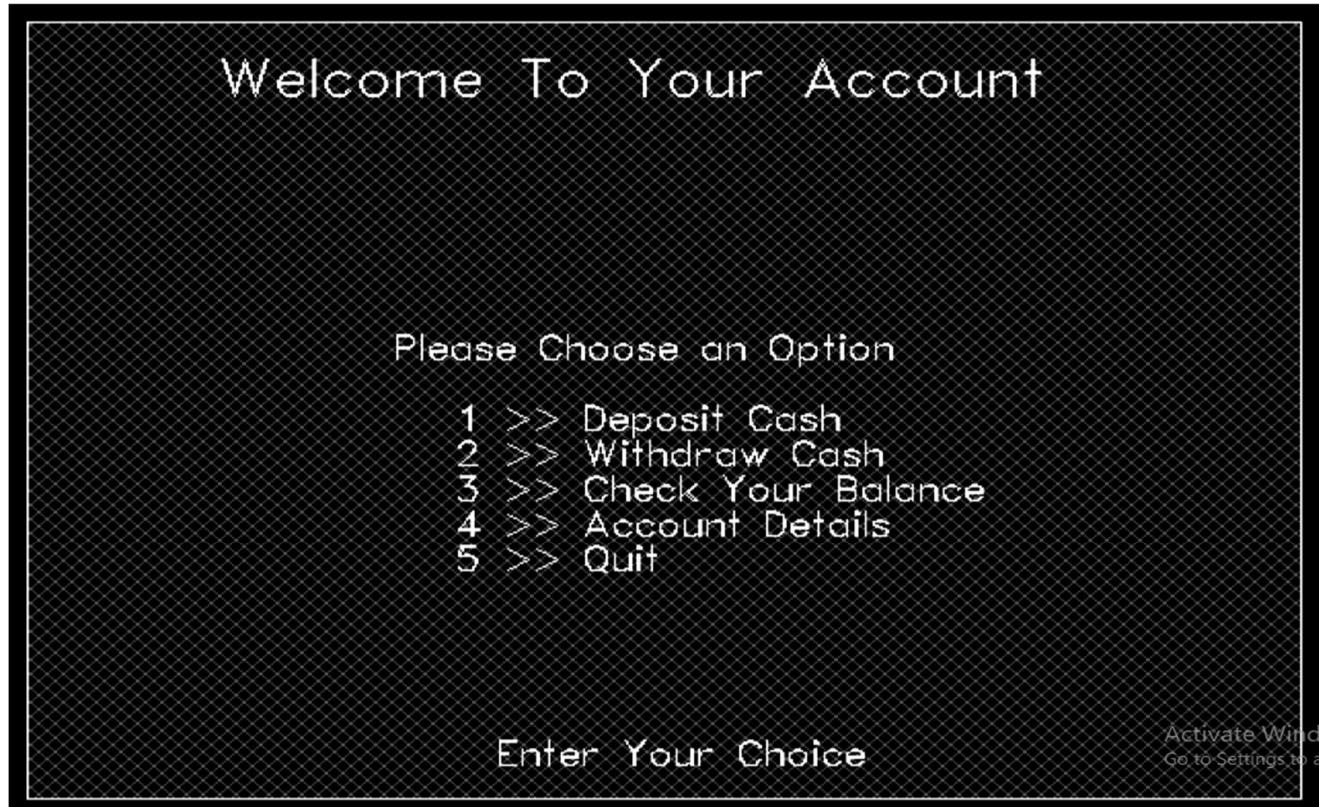


Algorithm Flowchart →

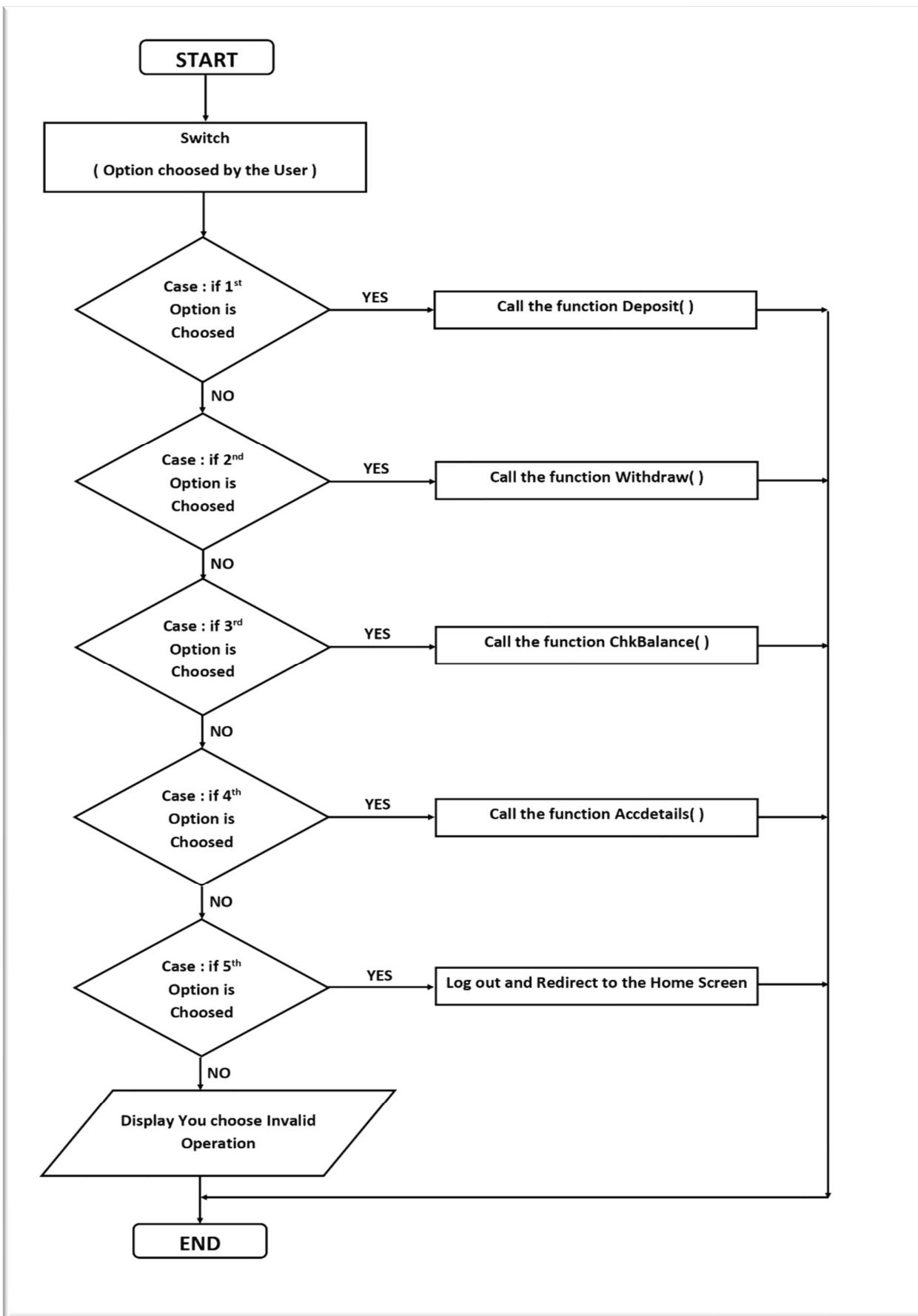


Transaction Options Screen →

This screen is the next screen after user successfully logs in into his account. Here, user has to enter his choice for the further proceedings. The flowchart explains in detail how this module works.



Algorithm Flowchart →

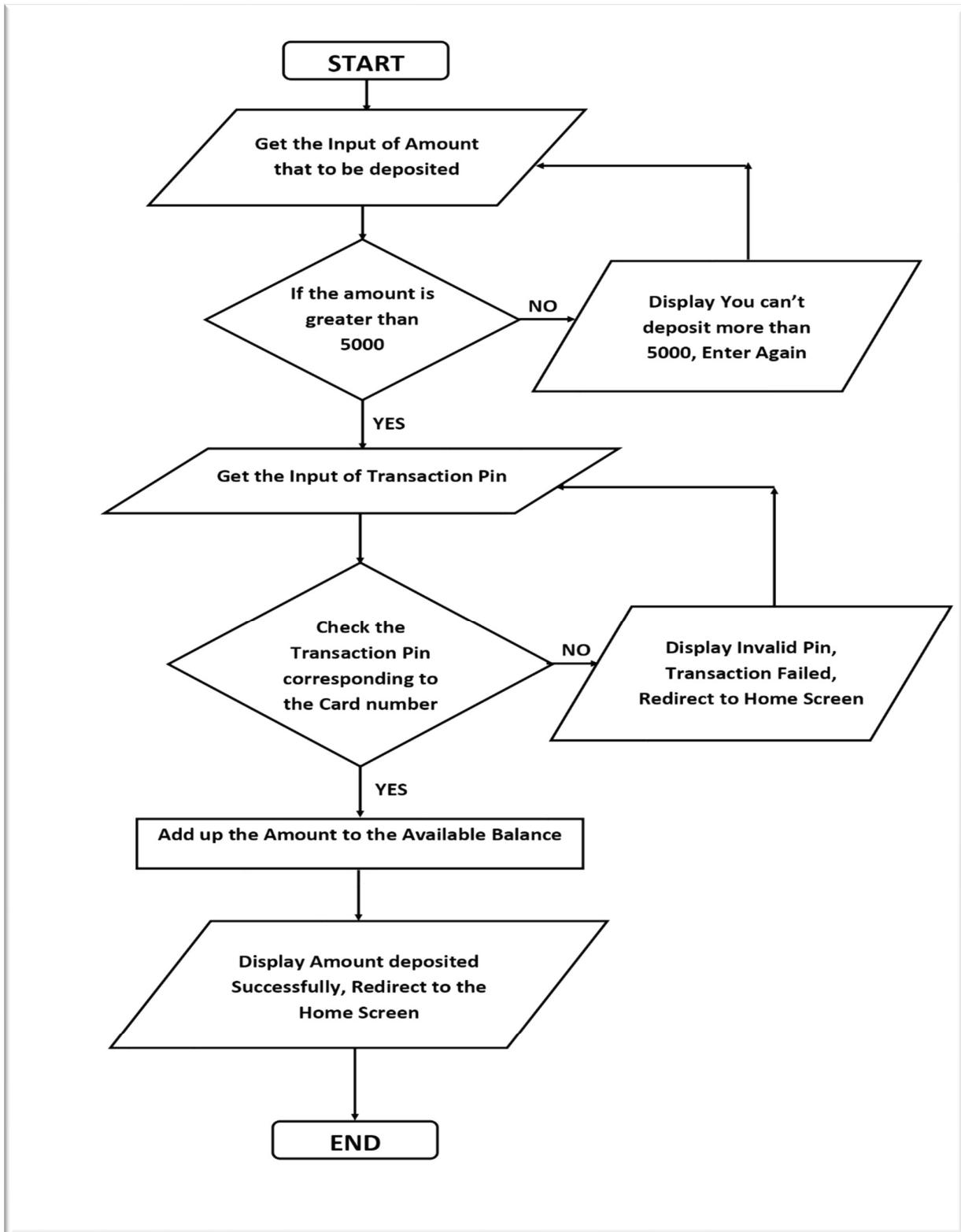


Deposit Screen →

This is the first option offered to the user. This module allows the user to deposit money in to his account. The working of this module is explained in detail in the flowchart.



Algorithm Flowchart →

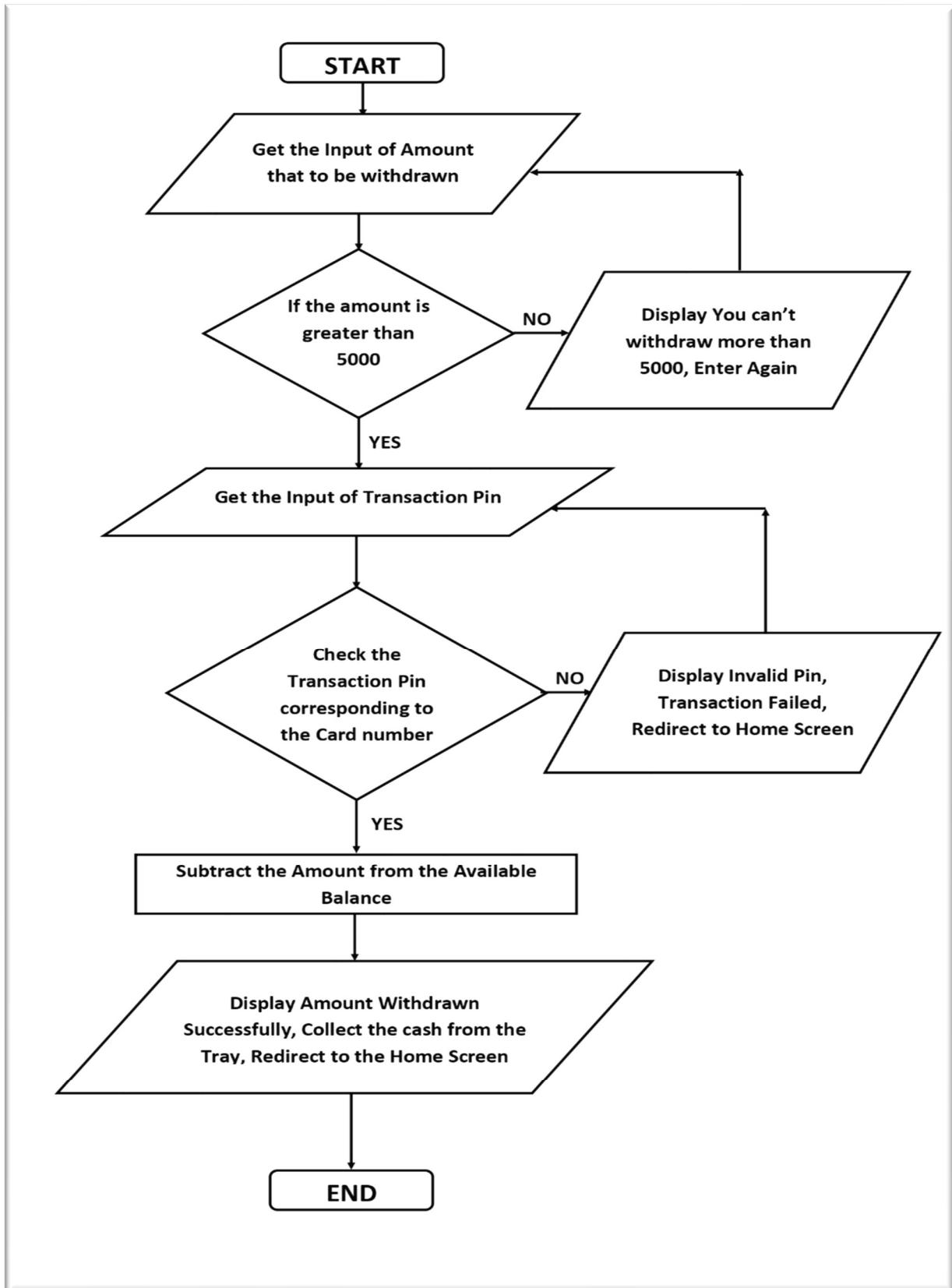


Withdraw Screen →

This is the second option presented to the user .This module works for the withdrawal of the cash from the user's account. The working of this module is explained well in the flowchart.

```
***** WITHDRAW CASH *****  
Amount should be in multiples of 100  
Enter Amount to be withdrawn : 3000  
  
Enter Your Transaction Pin : ****  
  
***** TRANSACTION SUCCESSFUL *****  
***** AMOUNT WITHDRAWN SUCCESSFULLY *****_
```

Algorithm Flowchart →

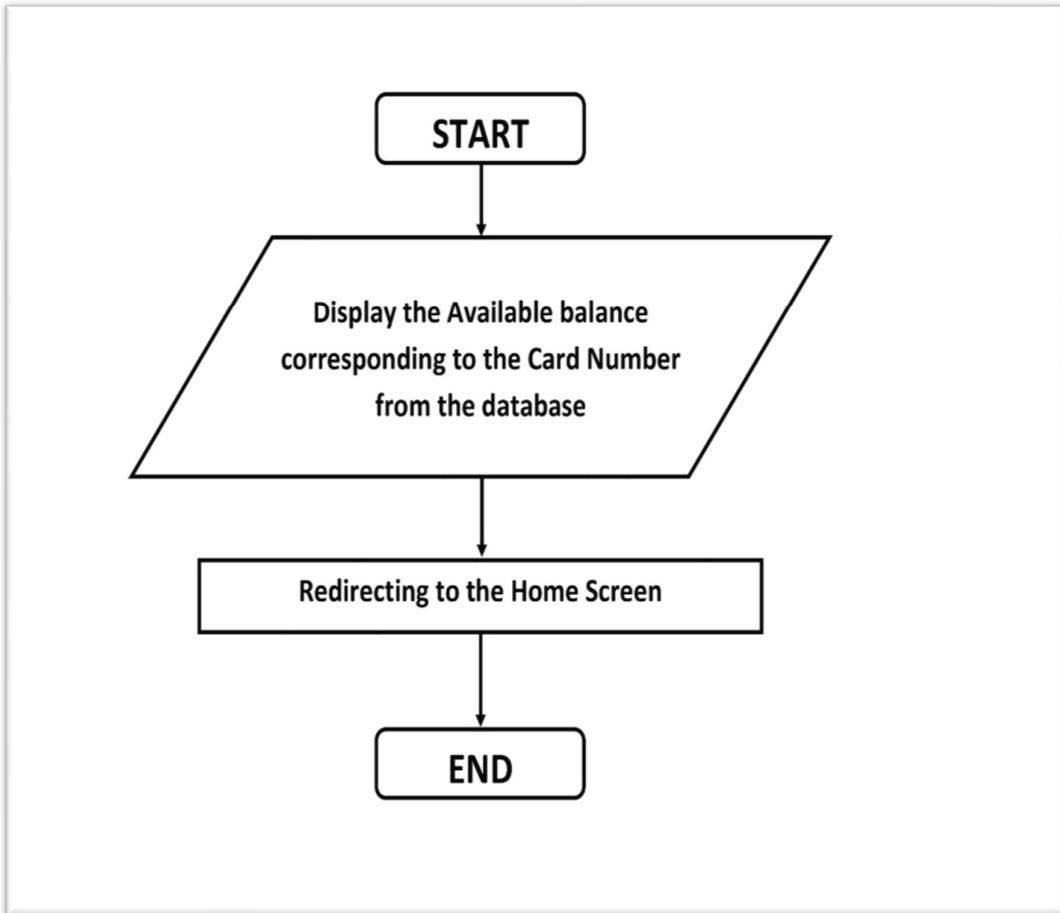


Check Balance Screen →

This is the third option presented to the user after he logs into his account. Using this module, user can view available balance in his account. The working of the module is represented on the flowchart.



Algorithm Flowchart →

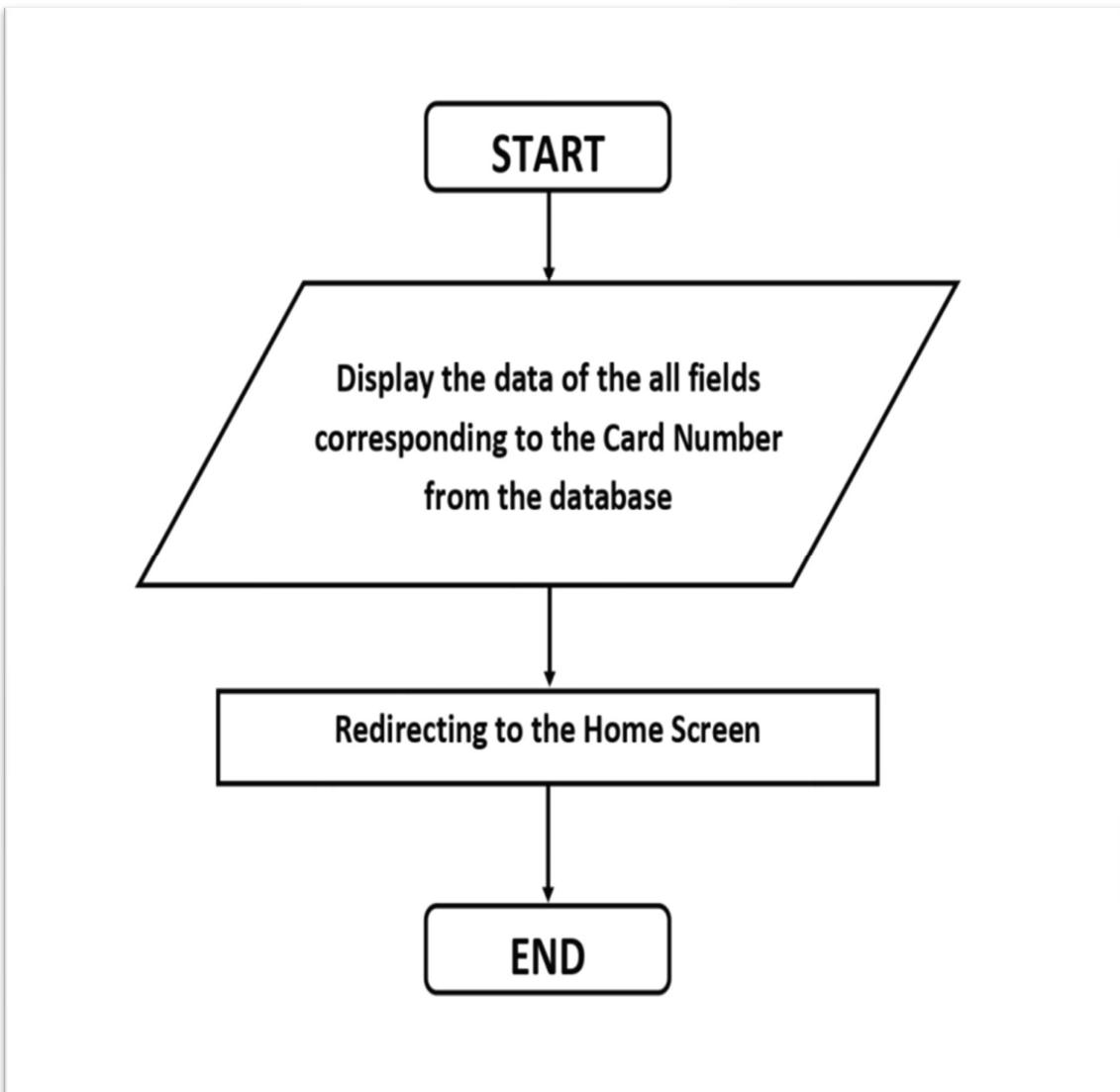


ACCOUNT DETAILS SCREEN →

This is the fourth option offered to the user. Using this, a user can view basic details related to the account on the go. The flowchart represents the working of this module.



Algorithm Flowchart →



Thank You Screen →

This is the final screen a user sees, after he completes all his desired processes. When the user completes his transaction successfully, a screen displaying Thank You is displayed for few seconds and then is redirected to the Home screen.



Source Code

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>           // All the Libraries needed for Implementation
#include <stdlib.h>
#include <string.h>

                           // All the functions to perform the particular operation

void Acdata();             // Function for Data Storage
int Login();                // Function for Login Mechanism
int Deposit();              // Function for mechanism of Cash Deposition
int Withdraw();             // Function for mechanism of Cash Withdraw
int Chkbalance();           // Function for mechanism of Checking Balance
int Accdetails();            // Function for mechanism of displaying Account Details
void Thanks();               // Function for Thank You Screen

typedef struct
{
    char Acno[13];
    char Acname[15];
    float Amt;
    char Cdno[13];           // Data of the User is stored in the form of these Structure
    int lpin;
    int tpin;
} Account;

Account *accs;
int gd = DETECT, gm;
int lpass=0, tpass=0, idx;

void main()
{
    int lck, tch;
    char ec;
    accs = (Account*)calloc(3, sizeof(Account));           /* Dynamic Memory Allocation for
                                                               Structure*/
    if(accs != NULL)
    {
        Acdata();
    }

start:
initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
rectangle(10,10,630,450);
setfillstyle(8,8);
floodfill(15,15,15);                                /* Graphics Code for Home Screen
                                                       Layout*/
settextstyle(3,0,4);
outtextxy(135,20,"CARDLESS ATM SERVICE");
outtextxy(210,180,"Hello Customer");
outtextxy(245,220,"Welcome :)");
settextstyle(3,0,2);
outtextxy(210,410,"Press Enter To Continue");
ec = getch();
closegraph();

if(ec == '5')
{
    exit(0);                                         // Exit mechanism for Administrator
```

```

}

lck = Login();
if(lck == 0)
{
    goto start;
}

clrscr();
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
rectangle(10,10,630,450);
setfillstyle(8,8);
floodfill(15,15,15);
settextstyle(3,0,4);
outtextxy(105,20,"Welcome To Your Account");
settextstyle(3,0,2);
outtextxy(190,180,"Please Choose an Option");
outtextxy(220,220,"1 >> Deposit Cash");
outtextxy(220,240,"2 >> Withdraw Cash");
outtextxy(220,260,"3 >> Check Your Balance");
outtextxy(220,280,"4 >> Account Details");
outtextxy(220,300,"5 >> Quit");
outtextxy(240,410,"Enter Your Choice");

tch = getch();
closegraph();
clrscr();

switch(tch)                                // Choice mechanism for choosing Transaction
{
    case '1':
        Deposit();
        break;

    case '2':
        Withdraw();
        break;

    case '3':
        Chkbalance();
        break;

    case '4':
        Accdetails();
        break;

    case '5':
        clrscr();
        Thanks();
        goto start;

    default:
        clrscr();
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        rectangle(10,10,630,450);
        setfillstyle(8,8);
        floodfill(15,15,15);
        settextstyle(3,0,4);
        outtextxy(250,200,"Invalid Choice");
        getch();
        closegraph();

        Thanks();
        clrscr();
}
/* Graphics Code for
Transaction Options Screen */

/* Code for resolving if choice
   is invalid */

```

```

        break;
    }

    goto start;

}

void Acdata()
{
    strcpy((accs+0)->Acno,"918602501968");
    strcpy((accs+0)->Acname,"Jayesh Bansal");
    (accs+0)->Amt = 16328.75;
    strcpy((accs+0)->Cdno,"973474010816");
    (accs+0)->lpin = 3927;
    (accs+0)->tpin = 1479;

    strcpy((accs+1)->Acno,"919827935115");
    strcpy((accs+1)->Acname,"Aakash Jain");
    (accs+1)->Amt = 27529.41;
    strcpy((accs+1)->Cdno,"983557353733");
    (accs+1)->lpin = 5735;
    (accs+1)->tpin = 3123;

    strcpy((accs+2)->Acno,"916232109001");
    strcpy((accs+2)->Acname,"Jitendra Singh");
    (accs+2)->Amt = 32920.60;
    strcpy((accs+2)->Cdno,"977374856631");
    (accs+2)->lpin = 1924;
    (accs+2)->tpin = 2437;
}

int Login()
{
    int i, j, ch, ac=0, pc=0, count;
    char c, *edcn;
    clrscr();
    printf("\n\n");
    printf("\t ***** LOGIN PAGE *****\n");
    printf("\n\n\n");
    printf("\t\t Enter Your ATM Card Number : ");

    card:
    count = 0;
    scanf("%s", edcn);

    for(i=0; i<3; i++)
    {
        if(strcmp(edcn, (accs+i)->Cdno) == 0)
        {
            printf("\n\n\t Enter Your Login Pin : ");
            for( j=0; j<4 ; j++)
            {
                c = getch();
                ch = c - '0';
                lpass = lpass*10 + ch;
                printf("*");
            }
            getch();
            idx = i;
            goto pass;
        }
    }
}

```

```

        else
        {
            count++;
        }
    }

if(count == 3)
{
    if(ac == 1)
    {
        printf("\n\n\n");
        printf("\t ***** LOGIN FAILED *****");
        getch();
        Thanks();
        clrscr();
        return 0;
    }

    gotoxy(1,6);
    delline();
    ac++;
    printf("\t\t Card Number not exist, Press Enter");
    getch();
    printf("\r\t\t Enter Your ATM Card Number Again : ");
    goto card;
}

pass:
if(lpass == (accs+idx)->lpin)
{
    printf("\n\n\n");
    printf("\t ***** LOGIN SUCCESSFUL *****");
    getch();
    clrscr();
    lpass = 0;
    return 1;
}

else
{
    if(pc == 1)
    {
        printf("\n\n\n");
        printf("\t ***** LOGIN FAILED *****");
        getch();
        Thanks();
        clrscr();
        return 0;
    }

    gotoxy(1,8);
    delline();
    pc++;
    printf("\t\t Incorrect Pin, Press Enter ");
    getch();
    lpass = 0;
    printf("\r\t\t Enter Your Login Pin Again : ");

    for( j=0; j<4 ; j++)
    {
        c = getch();

```

```

        ch = c - '0';
        lpass = lpass*10 + ch;
        printf("*");
    }
    getch();
    goto pass;
}
}

int Deposit()
{
    int i, ch, dac=0, dtc=0, damt;
    char c;

    clrscr();
    printf("\n\n");
    printf("\t ***** DEPOSIT CASH *****");
    printf("\n\n");
    printf("\t\t Amount should be in multiples of 100");
    printf("\n\n");
    printf("\t\t Enter Amount to be deposited : ");

    deposit:
    scanf("%d", &damt);

    if( damt%100 == 0 && damt < 5000 )
    {
        printf("\n\n");
        printf("\t\t Enter Your Transaction Pin : ");

        for( i=0; i<4 ; i++)
        {
            c = getch();
            ch = c - '0';
            tpass = tpass*10 + ch;
            printf("*");
        }
        getch();
        goto Transacpin;
    }

    else
    {
        if(dac == 1)
        {
            printf("\n\n");
            printf("\t ***** TRANSACTION FAILED *****");
            getch();
            Thanks();
            clrscr();
            return 0;
        }
        gotoxy(1,7);
        delline();
        dac++;
        printf("\t\t Invalid Amount, Press Enter");
        getch();
        printf("\r\t\t Enter Amount to be deposited Again : ");
        goto deposit;
    }
}

Transacpin:

if(tpass == (accs+idx)->tpin)

```

```

    {
        printf("\n\n\n");
        printf("\t ***** TRANSACTION SUCCESSFUL *****");
        (accs+idx)->Amt = (accs+idx)->Amt + (float)damt;
        printf("\n\t ***** AMOUNT DEPOSITED SUCCESSFULLY *****");
        getch();
        Thanks();
        clrscr();
        tpass = 0;
        return 1;
    }

else
{
    if(dtc == 1)
    {
        printf("\n\n\n");
        printf("\t ***** TRANSACTION FAILED *****");
        getch();
        Thanks();
        clrscr();
        return 0;
    }
    gotoxy(1,10);
    delline();
    dtc++;
    printf("\t\t Incorrect Pin, Press Enter");
    getch();
    tpass = 0;
    printf("\r\t\t Enter Your Transaction Pin Again : ");

    for( i=0; i<4 ; i++)
    {
        c = getch();
        ch = c - '0';
        tpass = tpass*10 + ch;
        printf("*");
    }
    getch();
    goto Transacpin;
}
}

int Withdraw()
{
    int i, ch, wac=0, wtc=0, wamt;
    char c;

    clrscr();
    printf("\n\n");
    printf("\t ***** WITHDRAW CASH *****");
    printf("\n\n");
    printf("\t\t Amount should be in multiples of 100");
    printf("\n\n");
    printf("\t\t Enter Amount to be withdrawn : ");

withdraw:
    scanf("%d", &wamt);

    if( wamt%100 == 0 && wamt < 5000 )
    {
        printf("\n\n");
        printf("\t\t Enter Your Transaction Pin : ");

        for( i=0; i<4 ; i++)

```

```

    {
        c = getch();
        ch = c - '0';
        tpass = tpass*10 + ch;
        printf("*");
    }
    getch();
    goto Transacpin;
}

else
{
    if(wac == 1)
    {
        printf("\n\n\n");
        printf("\t ***** TRANSACTION FAILED *****");
        getch();
        Thanks();
        clrscr();
        return 0;
    }
    gotoxy(1,7);
    delline();
    wac++;
    printf("\t\t Invalid Amount, Press Enter");
    getch();
    printf("\r\t\t Enter Amount to be withdrawn Again : ");
    goto withdraw;
}

```

Transacpin:

```

if(tpass == (accs+idx)->tpin)
{
    printf("\n\n\n");
    printf("\t ***** TRANSACTION SUCCESSFUL *****");
    (accs+idx)->Amt = (accs+idx)->Amt - (float)wamt;
    printf("\n\t ***** AMOUNT WITHDRAWN SUCCESSFULLY *****");
    getch();
    Thanks();
    clrscr();
    tpass = 0;
    return 1;
}

else
{
    if(wtc == 1)
    {
        printf("\n\n\n");
        printf("\t ***** TRANSACTION FAILED *****");
        getch();
        Thanks();
        clrscr();
        return 0;
    }

    gotoxy(1,10);
    delline();
    wtc++;
    printf("\t\t Incorrect Pin, Press Enter");
    getch();
    tpass = 0;
    printf("\r\t\t Enter Your Transaction Pin Again : ");

    for( i=0; i<4 ; i++)

```

```

        {
            c = getch();
            ch = c - '0';
            tpass = tpass*10 + ch;
            printf("*");
        }
        getch();
        goto Transacpin;
    }
}

int Chkbalance()
{
    int i, ch, ctc=0;
    char c, bal[40];

    clrscr();
    sprintf(bal,"AVAILABLE BALANCE : %.2f ",(accs+idx)->Amt);
    printf("\n\n");
    printf("\t***** FOR CHECKING AVAILABLE BALANCE *****");
    printf("\n\n\n");
    printf("\t\t Enter Your Transaction Pin : ");

    Transacpin:
    tpass = 0;

    for( i=0; i<4 ; i++)
    {
        c = getch();
        ch = c - '0';
        tpass = tpass*10 + ch;
        printf("*");
    }
    getch();

    if(tpass == (accs+idx)->tpin)
    {
        clrscr();
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        rectangle(10,10,630,450);
        setfillstyle(8,8);
        floodfill(15,15,15);
        settextstyle(3,0,4);
        outtextxy(210,20,"Account Balance");
        settextstyle(3,0,3);
        outtextxy(140,200,bal);
        getch();
        closegraph();
        Thanks();
        clrscr();
        tpass = 0;
        return 1;
    }

    else
    {
        if(ctc == 1)
        {
            printf("\n\n\n");
            printf("\t***** TRANSACTION FAILED *****");
            getch();
            Thanks();
            clrscr();
            return 0;
        }
    }
}

```

```

        }

        gotoxy(1,5);
        delline();
        ctc++;
        printf("\t\t Incorrect Pin, Press Enter");
        getch();
        printf("\r\t\t Enter Your Transaction Pin Again : ");
        goto Transacpin;
    }
}

int Accdetails()
{
    int i, ch, atc=0;
    char c, Ac[40], Cd[30], Aname[40], Balance[30];
    clrscr();
    sprintf(Ac,"Account Number : %s", (accs+idx)->Acno);
    sprintf(Cd,"Card Number : %s", (accs+idx)->Cdno);
    sprintf(Aname,"Account Holder Name : %s", (accs+idx)->Acname);
    sprintf(Balance,"Available Balance : %.2f", (accs+idx)->Amt);

    printf("\n\n");
    printf("\t ***** FOR ACCOUNT DETAILS *****");
    printf("\n\n\n");
    printf("\t\t Enter Your Transaction Pin : ");

Transacpin:
tpass = 0;

    for( i=0; i<4 ; i++)
    {
        c = getch();
        ch = c - '0';
        tpass = tpass*10 + ch;
        printf("*");
    }
    getch();

    if(tpass == (accs+idx)->tpin)
    {
        clrscr();
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        rectangle(10,10,630,450);
        setfillstyle(8,8);
        floodfill(15,15,15);
        settextstyle(3,0,4);
        outtextxy(210,20,"Account Details");
        settextstyle(3,0,3);
        outtextxy(130,200,Ac);
        outtextxy(130,230,Cd);
        outtextxy(130,260,Aname);
        outtextxy(130,290,Balance);
        getch();
        closegraph();
        Thanks();
        clrscr();
        tpass = 0;
        return 1;
    }

else
{
    if(atc == 1)

```

```

    {
        printf("\n\n\n");
        printf("\t ***** TRANSACTION FAILED *****\n");
        getch();
        Thanks();
        clrscr();
        return 0;
    }

    gotoxy(1,5);
    delline();
    atc++;
    printf("\t\t Incorrect Pin, Press Enter");
    getch();
    printf("\r\t\t Enter Your Transaction Pin Again : ");
    goto Transacpin;
}
}

void Thanks()
{
    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
    rectangle(10,10,630,450);
    setfillstyle(8,8);
    floodfill(15,15,15);
    settextstyle(3,0,4);
    outtextxy(210,200," THANK YOU! ");
    delay(5000);
    closegraph();
}

```

ADVANTAGES

After studying the software in detail, let's look into the some of the advantages that this software provides us.

1. It requires less effort to complete the transactions.
2. It reduces the time consumed considerably.
3. There is no need to maintain the bulk of papers for data management.
4. There is an added level of security – Transaction pin. Thus, enhancement into user security.
5. The user does not need to carry ATM card for small transactions.
6. It enhances user experience and user convenience.
7. The user can verify details about his account and keep an eye on the account balance on the go.

CONCLUSION

Therefore, our project provides a way in which this problem can be solved. People will be able to perform transactions only for small amounts.

This way people will have to worry less about carrying ATM cards. This will reduce the time consumed in the process. There will be an added advantage in the form of transaction pin which is a option for double security.

REFERENCES

- <https://www.javatpoint.com/c-programming-language-tutorial>
- <https://www.geeksforgeeks.org/c-programming-language/>
- <https://www.slideshare.net/khalidbazgamah/atm-project-75361416>
- <https://www.sanfoundry.com/c-program-atm-transaction/>