**Name: 1)** Vasanthakumari Issac Nadar **Roll no:** 26

2) Ruchita Patil 59

3) Sheetal Mane 53

4) Swapnali Shinde 57

**Course:** BCA SEM 2

**Subject:** Minor project Ist

**Institution Name**: Bharati Vidyapeeth Institution of Management (Navi Mumbai)

**Abstract**

**Abstract**

The project banking system includes bank enquiry, deposit, withdraw and transfer of money from bank accounts. In starting stage of banking all the records are maintained in books/journals etc. It had drawbacks like lost of transactions, time consuming, lots of human efforts, etc. The term E-Banking had changed the method of running banks. It had reduced the human efforts and time. The traction is maintained easily and errors are easily rectified.

The project is to run the bank in easy and efficient manner. It is user friendly, even a small child can also use it easily. It includes all the main work done in a bank. Less human effort is required to maintain the transactions. The project aims at creation of squire internet banking. The exciting part of this project is; it displays Transaction reports, Statistical Summary of Account type.

The domain “banking system” keeps the day by day tally records as a complete banking. It can keep the information of account type, account opening form.

A bank is a business; banks sell financial services such as Vehicle loans, home mortgage loans, business loans, checking accounts, credit card services, certificates of deposit, and individual retirement accounts etc...  
Some people go to banks in search of a safe place to keep their money. Others are seeking to borrow money to buy a house or a car, start a business, expand a farm, pay for college, or do other things that require borrowing money.

Where do banks get the money to lend?   
They get it from people who open accounts. Banks act as go-betweens for people who save and people who want to borrow. If savers didn’t put their money in banks, the banks would have little or no money to lend.

The objective of this project is to help the customer of the bank to do their account transactions easily. To show how the banking system works and what are the features of banking system. This system acts as a standard interface between the client and the bank.

To develop and implement new ideas and innovations in banking services, operations and procedures.  
To project a good public image of banking as a service industry and develop good public   
relations.

**Algorithm**

**Algorithm:**

Start.

Read name.

Read account number.

Print choices 1: bank enquiry 2:deposit 3:withdraw 4:transfer.

Read choice.

If 1, display balance.

Print balance.

Print do you want to continue?

Read choice 1: yes or 2: no.

If 1, continue.

Print choices 1: bank enquiry 2: deposit 3: withdraw 4: transfer.

If 2, deposit.

Read the amount you want deposit.

Print the amount you had deposited with a message that the amount is deposited.

1 to continue.

Print choices 1: bank enquiry 2: deposit 3: withdraw 4: transfer.

If 3: withdraw.

Read the amount you want to withdraw.

Print the amount you had withdrawn with a message that the amount is withdrawn.

Press 1 to continue.

Print choices 1: bank enquiry 2: deposit 3: withdraw 4: transfer.

If 4: transfer.

Read the account number to which you want to transfer the money.

Read the amount to be transferred.

Print that the amount transferred.

Enter 2 to stop.

Print the transactions.

End.

Bank Enquiry Algorithm:

Start.

Read name

Read account number

Read the balance from system

Print balance

Print Transaction

End

Deposit Algorithm:

Start.

Read name.

Read account number.

Print deposit.

Read the amount you want deposit.

Print the amount you had deposited with a message that the amount is deposited.

End.

Withdraw Algorithm:

Start.

Read name.

Read account number.

Print withdraw

Read the amount you want to withdraw.

Print the amount you had withdrawn with a message that the amount is withdrawn.

Display bill with the details of transactions.

End.

Transfer Algorithm:

Start.

Read name.

Read account number.

Print transfer.

Read the account number to which you want to transfer the money.

Read the amount to be transferred.

Print that the amount transferred.

Print the transactions.

End.

**Flowchart**

**Flowchart**

Y

Y

Y

N

N

N

N

Y

Fig 1.1

If account no valid

Print transaction

Check balance

Display balance

Bank enquiry

a

Read name

numbar

Read account no

Fig 1.2 Balance Enquiry

Read name

Read account no.

Deposit

Read amt to deposit

Print the deposit amt

Fig 1.3 Deposit

Read name

numbar

Read account no

Withdraw

Read amt to withdraw

a

Print the withdraw amt

Display the transaction

Fig 1.4 Withdraw

START

Read Name

Read Account No

Print Transfer

Read a/c no to transfer amount

Read amount

Print amount transferred

END

.

Fig 1.5 Transfer

**Requirement**

**Requirement:**

Hardware:

800MHz processor or above.,20 MB of hard disk space.

Software:

Windows 95 or higher operating system, Turbo c.

**Description:**

**Turbo C** is a discontinued Integrated Development Environment and compiler for the C programming language from Borland. First introduced in 1987, it was noted for its integrated development environment, small size, fast compile speed, comprehensive manuals and low price.

In May 1990, Borland replaced Turbo C with Turbo C++. In 2006, Borland reintroduced the **Turbo** moniker.

**Version 1.0** (May 13, 1987) offered the first integrated development environment for C on IBM PCs. Like many Borland products of the time, the software was bought from another company (in this case **Wizard C** by Bob Jervis[[2]](https://en.wikipedia.org/wiki/Borland_Turbo_C" \l "cite_note-2)), and branded with the "Turbo" name

**Version 1.5** (January 1988) was an incremental improvement over version 1.0. It included more sample programs, improved manuals and bug fixes. It was shipped on five 360 KB diskettes of uncompressed files, and came with sample C programs, including a stripped down spreadsheet called [mcalc](https://en.wikipedia.org/w/index.php?title=Mcalc&action=edit&redlink=1" \o "Mcalc (page does not exist)).

**Version 2.0** (late 1988) featured the first "blue screen" version, which would be typical of all future Borland releases for MS-DOS. The American release did not have Turbo Assembler or a separate debugger.

**Turbo C++ 3.0** was released in 1991 (shipping on November 20), and came in amidst expectations of the coming release of Turbo C++ for Microsoft Windows. Initially released as an MS-DOS compiler, 3.0 supported C++ templates, Borland's inline assembler, and generation of MS-DOS mode executables for both 8086 real mode and 286 protected mode (as well as the Intel 80186.) 3.0 implemented AT&T C++ 2.1, the most recent at the time. The separate Turbo Assembler product was no longer included, but the inline-assembler could stand in as a reduced functionality version.

**coding**

**Coding:**

#include<stdio.h>

#include<conio.h>

int main()

{

float draw,dep,transfer;

char name[24];

float balance=5000;

int account1,account,pin;

int b,type;

int abc,option,zyx;

int transaction=1;

clrscr();

printf("\n\t\t\t\tBANKING SYSTEM\n\n");

printf("");

printf("\t\t\4\t\t\t\t\t\t\4\n");

printf("\t\t\4\t\t\t\t\t\t\4\n");

printf("\t\t\4\tWELCEME TO BHRAMA BANK LTD\t\t\4\n");

printf("\t\t\4\t\t\t\t\t\t\4\n");

printf("\t\t\4\t\t\t\t\t\t\4\n");

printf("\t\t\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4\4");

printf("\nEnter your name:");

scanf("%s",&name);

printf("Enter your account number:");

scanf("%d",&type);

while(transaction==1)

{

abc:

//int option;

printf("\nChoose what you want to do\n");

printf("1-Balance Enquiry\n");

printf("2-Deposit\n");

printf("3-Withdraw\n");

printf("4-Transfer\n");

fflush(stdin);

scanf("%d",&option);

switch(option)

{

case 1:

printf("\t\t\t\*BALANCE ENQUIRY\*\n\n");

printf("your current balance is:%.2fRs\n",balance);

break;

case 2:

printf("\n\t\t\t\*DEPOSIT AMOUNT\*\n\n");

printf("how much money do you want to deposit:");

scanf("%f",&dep);

if(dep>0 && dep<=20000)

{

printf("\nyour %.2fRs deposited in your account \n\n",dep);

balance+=dep;

}

else if(dep>20000)

{

printf("\nyou cant deposit that much amount in one time\n0\n");

}

else

{

printf("\ninvalid deposit amount\n");

}

break;

case 3:

printf("\n\t\t\*WITHDRAW AMOUNT\*\n\n");

printf("how much money do you want to withdraw:");

scanf("%f",&draw);

if(draw<=balance && draw<=20000)

{

printf("\nyou just withdraw %.2fRs\n\n",draw);

balance-=draw;

}

else if(draw>20000)

{

printf("\nyou dont have enfough moneu\n\n");

}

break;

case 4:

printf("\t\t\t\*TRANSFER AMOUNT\*\n\n");

printf("\n\t\tAccount you want to transfer:");

scanf("%d",&account1);

printf("how much amount?: ");

scanf("%f",&transfer);

if(balance>=transfer)

{

printf("\nyour %.2fRs successfully transfered\n\n",transfer);

balance-=transfer;

}

else

{

printf("\nyou do not have sufficient balance\n\n");

}

break;

default:

printf("inalid transaction\n");

}

transaction=1;

//while (transaction!=1 && transaction!=2)

{

printf("do you want to do another transaction?\n");

printf("1: yes 2: no\n");

scanf("%d",&b);

if(b==1)

{

goto abc;

}

if(b==2)

{

goto zyx;

}

zyx:

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

printf("\n\t\t\t\tBHARAMA BANK LIMITED\n");

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

printf("\t\t\tDate:19/02/2017 Time:8:20 pm\n");

printf("\n\n\t\t\t Your name: %s\n\n",name);

printf("\t\t\t Your account no: %d\n\n",type);

if(dep>=0 && dep<20000)

{

printf("\t\t\t\4You've deposited %.2fRs\n",dep);

printf("\t\t\t\4You've deposited 0Rs\n");

}

if(draw>0 && draw<=20000 && draw<=balance)

{

printf("\t\t\t\4You've withdrawn %.2fRs\n",draw);

}

else

{

printf("\t\t\t\4You've tranfered %.2fRs\n");

}

printf("\n\t\t\t\t Thank ypu! \n");

printf("\t\t\t welceme to BHARAMA BANK LIMITED\n");

printf("\t\t\t www.BHARAMABANK.LTD.com\n");

getch();

exit(0);

}

}

**screenshot**

**Screen short:**

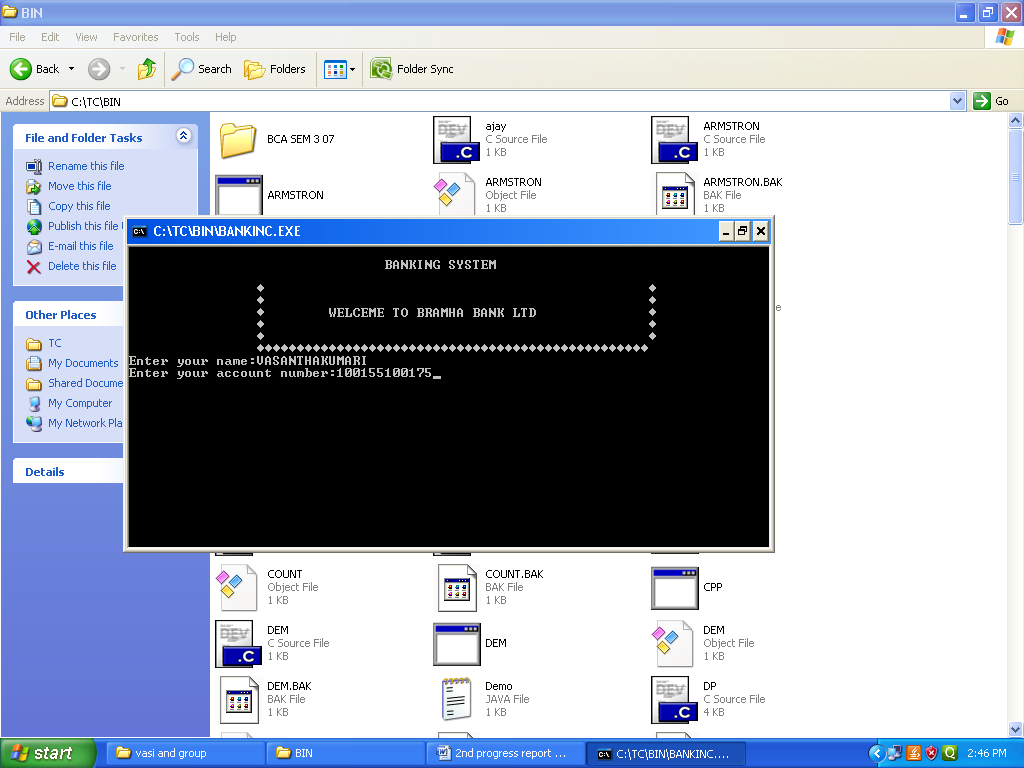


Fig 2.1

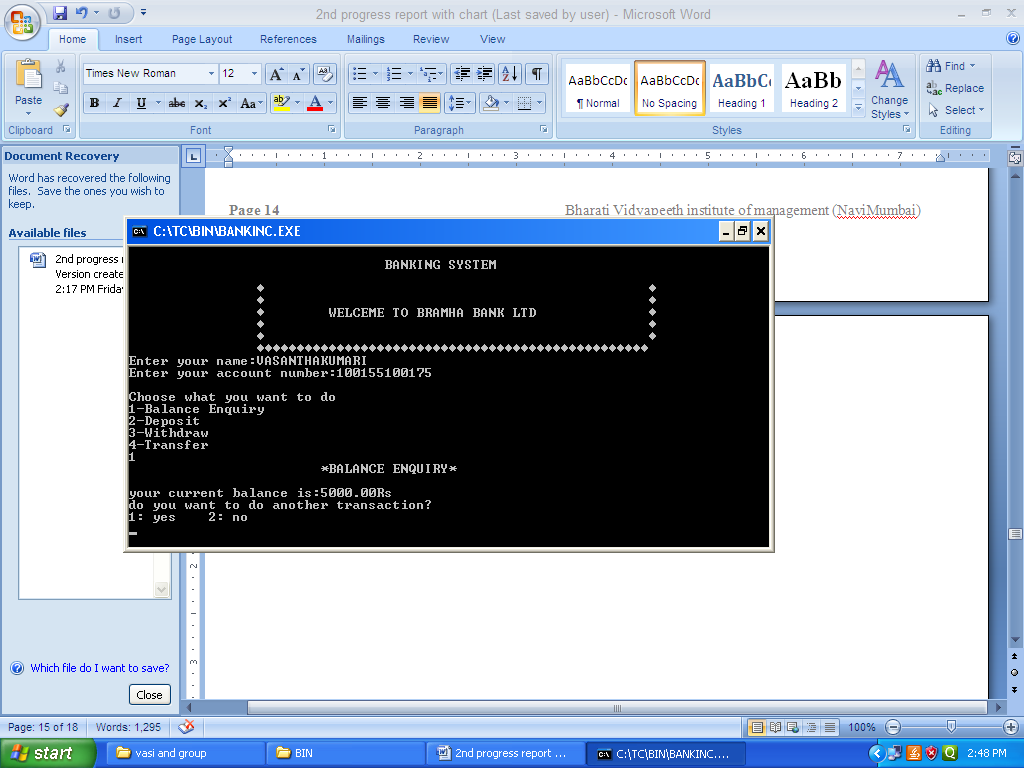


Fig 2.2 Balance Enquiry

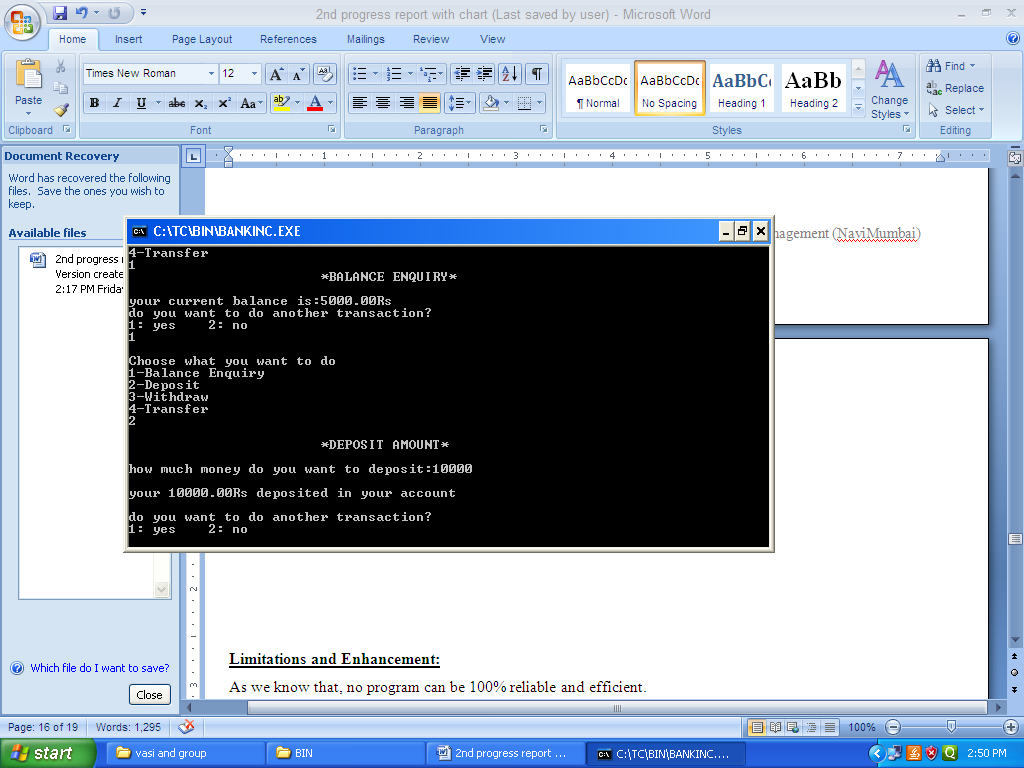


Fig 2.3 Deposit Amount

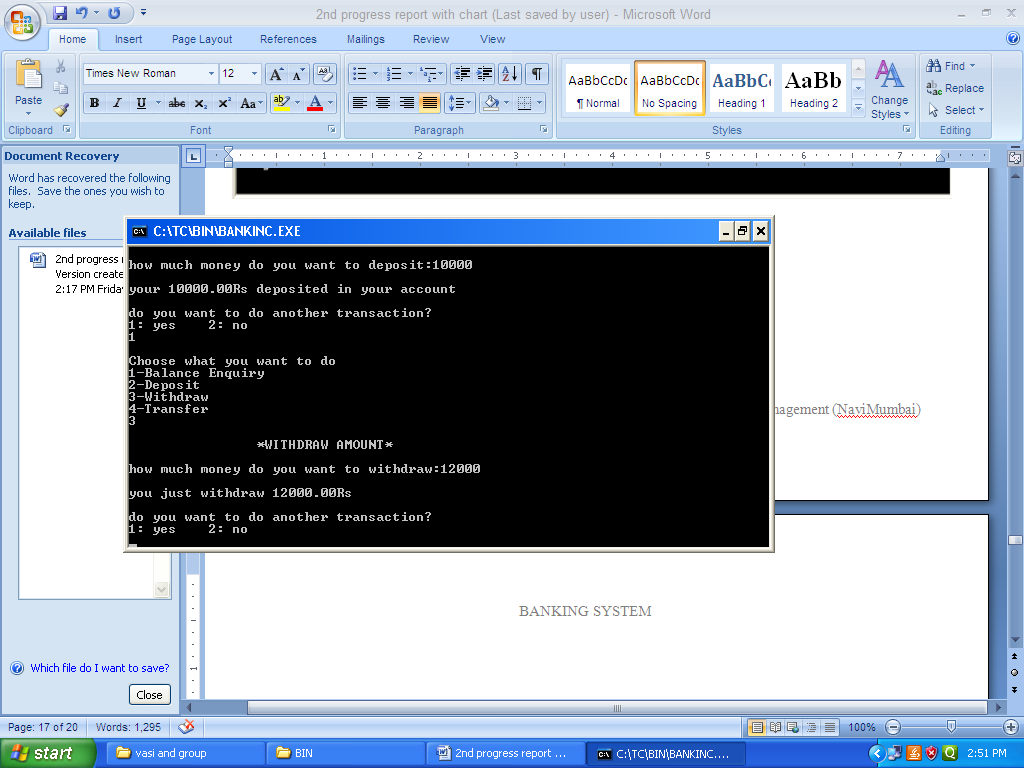


Fig 2.4 Withdraw

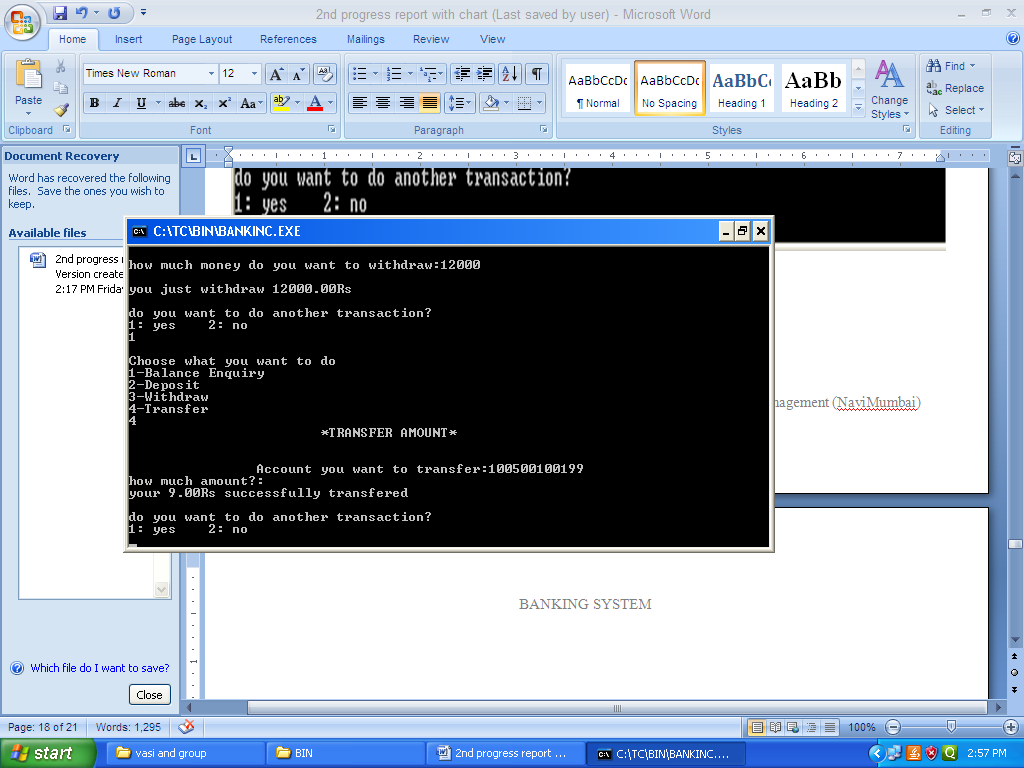


Fig 2.5 Transfer Amount

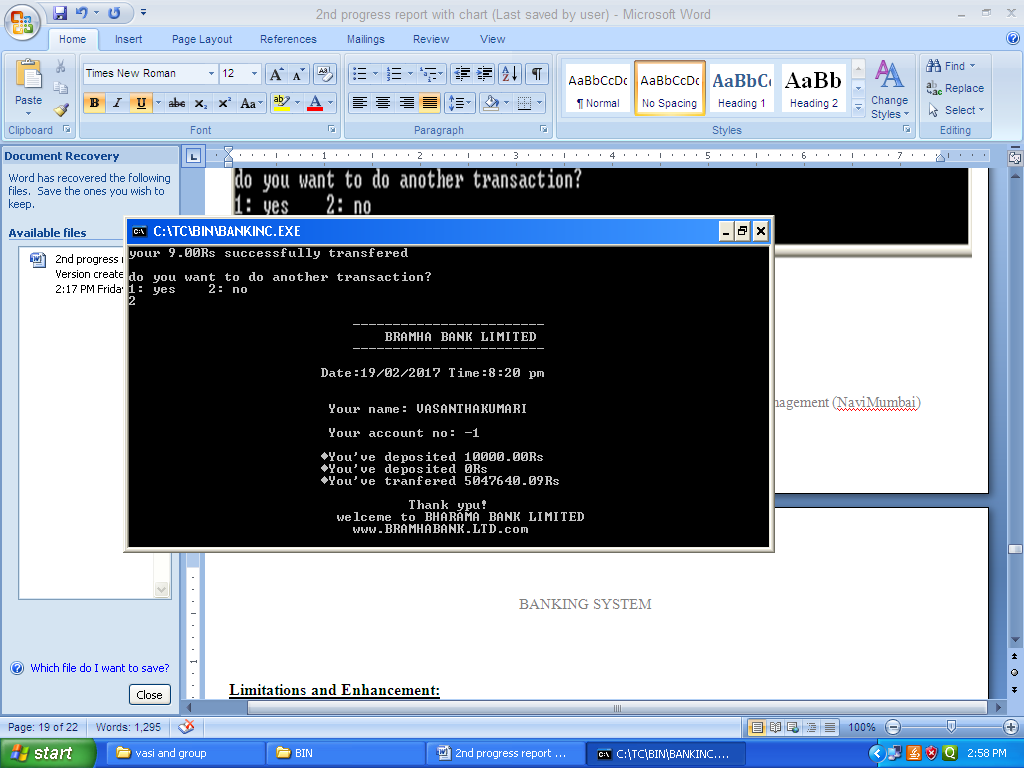
****

Fig 2.6 Transaction

**Limitations and Enhancement**

**Limitations and Enhancement:**

Limitations:

As we know that, no program can be 100% reliable and efficient.

There are some drawbacks in the proposed system which are as follows:

1. Database is required for the system.
2. Possibility of human errors.
3. Customer relationship is limited.
4. Limited use of technology.

Enhancement:-

1. Users can add extra enhancement in the system as per necessary in the future for fulfillment of their requirements.
2. For security purpose, advanced encryption techniques can be applied.

**Duration**

**Duration**

This project is based on banking system. The topic needs to be studied in depth first, before starting the project. We assume that the project will be completed within in 2 months, as the coding and documentation needs maximum time to complete.

**Gantt chart**

**Gantt chart:**

**References:-**

Bibliography:

* “Progranning in ANSI C”, Balaguruswamy ,6th edition, Tata McGraw-Hill Edication Pvt.Ltd,2012.
* “Let Us C”,Yashvant Kanetkar, 13th edition, BPB publication.

Webliography :-

* [**http://www.slideshare.net/nitishxavier11/synopsis-on-bank-management-system**](http://www.slideshare.net/nitishxavier11/synopsis-on-bank-management-system) ,
* [**http://projectsgeek.com/2013/06/banking-project-in-c-language-source-code.c**](http://projectsgeek.com/2013/06/banking-project-in-c-language-source-code.c) **prog** ,
* **http://study.com/academy/lesson/banking-system-definition-types,c ,**
* [**http://www.investopedia.com/university/banking-system/**](http://www.investopedia.com/university/banking-system/) **,**
* [**http://whatis.techtarget.com/definition/core-banking-system**](http://whatis.techtarget.com/definition/core-banking-system) **.**