**COMPUTER**

**PROJECT**



**BANK**

**MANAGEMENT**

MADE BY

Ananya Jain

XII B

**INDEX**

|  |  |  |
| --- | --- | --- |
| **S.No** | **CONTENTS** | **Page No.** |
|  | **Certificate** | 3 |
|  | **Acknowledgement** | 4 |
|  | **Introduction** | 5 |
|  | **-Objectives of Proposed System**  **-Advantages of Proposed System** | 6 |
|  | **Module chart** | 7 |
|  | **Header files and their purposes** | 8 |
|  | **Classes and functions used** | 9 |
|  | **Coding** | 10 |
|  | **Output** | 20 |
|  | **Requirements** | 29 |
|  | **Bibliography** | 30 |

**CERTIFICATE**

**This is to certify that**

**MISS ANANYA JAIN**

Of class **XII B**

has successfully completed the project report on

**BANK MANAGEMENT SYSTEM**

Under the guidance of Mrs. Suchanda Sen

for the fulfillment of Class 12 session ending examination for session 2018-2019.

(Suchanda Sen) PGT (CS)

**ACKNOWLEDGEMENT**

I have been supported by many people in completing my project report titled **Bank Management System.**

I am very grateful to Mrs. Suchanda Sen for her moral support and guidance.

I would like to thank Carmel Convent School and all its members for providing me this unique opportunity for self-learning.

I would also like to thank my incredibly supportive peer group for helping me out with the accomplishment of the Project.

Last but not the least I am very thankful to my worthy parents because without their incredible support, it would have been very difficult for me to do this project work successfully.

**INTRODUCTION**

A Bank Management software is essentially a software which which can undertake simple banking operations.

In earlier times before the advent of computers, Banks did exist but everything from maintaining records to computing lengthy and complex calculations was done manually. It had several inherent disadvantages, It not only required large manpower and efficient workers but also very slow and error prone.

However in recent times everything is done with the help of computers. There are a variety of software available today which can do this task efficiently, fast and accurately without requiring almost any sort of manual labour.

This software also aims to do the same.

**OBJECTIVES OF PROPOSED SYSTEM-**

* Act as an Basic Bank Management software
* Increases accuracy and efficiency in operations
* To reduce large amount of Paperwork
* To reduce manual labor

**ADVANTAGES OF PROPOSED SYSTEM-**

* No need of doing manual documentation
* Simple controls , higher knowledge of computers not required
* Efficient and Accurate
* Available free of cost

**MODULE CHART**

**Main main**

**New account write\_account()**

**Modify an account modify\_account(num)**

**Withdrawamount withdraw(num)**

**All account holder list display\_all()**

**Balance enquirydisplay\_sp(num)**

**Delete an account delete\_account(num)**

**exit**

**depositamount deposit(num)**

**HEADER FILES AND**

**THEIR PURPOSES**

* **fstream.h :**It is used to create files , write on files , and read from files.
* **ctype**.**h :**Header file for functions like isalpha() , isdigit() , toupper() , tolower() ,isupper() etc.
* **iomanip.h :** It defines functions like setw() , setprecision() etc.
* **conio.h :**This header file declares clrscr().
* **stdio.h :** It defines functions like gets , puts etc.
* **process.h :**This header file declares the function exit(0).

**CLASSES AND**

**FUNCTIONS USED**

Class - account

MEMBER FUNCTIONS:

**void create\_account()** - function to get data from user

**void show\_account()** - function to show data on screen

**void modify()** - function to get new data from user

**void dep(int)** - function to accept amount and add to balance amount

**void draw(int)** -function to accept amount and subtract from balance amount

**void report()** - function to show data in tabular format

**intretacno()** - function to return account number

**intretdeposit()** - function to return balance amount

**char rettype()** - function to return type of account

OTHER FUNCTIONS :

**void write\_account()** - function to write record in binary file

**void display\_sp(int)** - function to display account details given by user

**void modify\_account(int)** - function to modify record of file

**void delete\_account(int)** - function to delete record of file

**void display\_all()** - function to display all account details

**void deposit(int)** - function to desposit amount for given account

**void withdraw(int)** - function to withdraw amount for given account

**void intro()** - introductory screen function

**CODING**

#include<fstream.h>

#include<ctype.h>

#include<iomanip.h>

#include<conio.h>

#include<stdio.h>

#include<process.h>

class account

{

int acno;

char name[50];

int deposit;

char type;

public:

void create\_account();

void show\_account();

void modify();

void dep(int);

void draw(int);

void report();

int retacno();

int retdeposit();

char rettype();

};

void account::create\_account()

{

cout<<"\nEnter the account no. : ";

cin>>acno;

cout<<"\nEnter the name of the account holder : ";

gets(name);

cout<<"\nEnter type of the account (Current-c/Saving-s) : ";

cin>>type;

type=toupper(type);

cout<<"\nEnter the initial amount";

cout<<"\n(>=500 for Saving and >=1000 for current ) : ";

cin>>deposit;

cout<<"\n\n\nACCOUNTCREATED!..";

}

void account::show\_account()

{

cout<<"\nAccountno. : "<<acno;

cout<<"\nAccount holder's name : "<<name;

cout<<"\nType of account : "<<type;

cout<<"\nBalanceamount : "<<deposit;

}

void account::modify()

{

cout<<"\nAccount no."<<acno;

cout<<"\n\nEnter new name of the account holder : ";

gets(name);

cout<<"\nEnter new type (C/S) : ";

cin>>type;

type=toupper(type);

cout<<"\nEnter the changed amount : ";

cin>>deposit;

}

void account::dep(int x)

{

deposit+=x;

}

void account::draw(int x)

{

deposit-=x;

}

void account::report()

{

cout<<acno<<setw(10)<<" "<<name<<setw(9)<<" "<<type<<setw(11)<<deposit<<endl;

}

int account::retacno()

{

return acno;

}

int account::retdeposit()

{

return deposit;

}

char account::rettype()

{

return type;

}

//FUNCTION DECLARATIONS

void write\_account();

void display\_sp(int);

void modify\_account(int);

void delete\_account(int);

void display\_all();

void deposit(int);

void withdraw(int);

void intro();

// THE MAIN FUNCTION OF THE PROGRAM

int main()

{

char ch;

int num;

clrscr();

intro();

do

{

clrscr();

cout<<"\n\n\n\t\*\*\*MAIN MENU\*\*\*";

cout<<"\n\n\t1. NEW ACCOUNT";

cout<<"\n\n\t2. DEPOSIT AMOUNT";

cout<<"\n\n\t3. WITHDRAW AMOUNT";

cout<<"\n\n\t4. BALANCE ENQUIRY";

cout<<"\n\n\t5. ALL ACCOUNT HOLDER LIST";

cout<<"\n\n\t6. CLOSE AN ACCOUNT";

cout<<"\n\n\t7. MODIFY AN ACCOUNT";

cout<<"\n\n\t8. EXIT";

cout<<"\n\n\tSelect Your Option (1-8) : ";

cin>>ch;

clrscr();

switch(ch)

{

case '1':write\_account();

break;

case '2':cout<<"\n\nEnter the account no. : ";

cin>>num;

deposit(num);

break;

case '3':cout<<"\n\nEnter the account no. : ";

cin>>num;

withdraw(num);

break;

case '4':cout<<"\n\nEnter the account no. : ";

cin>>num;

display\_sp(num);

break;

case '5':display\_all();

break;

case '6':cout<<"\n\nEnter the account no. : ";

cin>>num;

delete\_account(num);

break;

case '7':cout<<"\n\nEnter the account no. : ";

cin>>num;

modify\_account(num);

break;

case '8':cout<<"\n\n\tThanks for using bank managemnt system";

break;

default :cout<<"\Enter correctly";

}

getch();

}while(ch!='8');

return 0;

}

// FUNCTION TO WRITE IN FILE

void write\_account()

{

account ac;

ofstream outFile;

outFile.open("acc.dat",ios::binary|ios::app);

ac.create\_account();

outFile.write((char \*) &ac, sizeof(account));

outFile.close();

}

// FUNCTION TO READ SPECIFIC RECORD FROM FILE

void display\_sp(int n)

{

account ac;

int flag=0;

ifstream inFile;

inFile.open("acc.dat",ios::binary);

if(!inFile)

{

cout<<"File could not be opened !! Press any Key...";

return;

}

cout<<"\nBALANCE DETAILS\n";

while(inFile.read((char \*) &ac, sizeof(account)))

{

if(ac.retacno()==n)

{

ac.show\_account();

flag=1;

}

}

inFile.close();

if(flag==0)

cout<<"\n\nAccount number does not exist";

}

// FUNCTION TO MODIFY RECORD OF FILE

void modify\_account(int n)

{

int found=0;

account ac;

fstream File;

File.open("acc.dat",ios::binary|ios::in|ios::out);

if(!File)

{

cout<<"File could not be opened !! Press any Key...";

return;

}

while(File.read((char \*) &ac, sizeof(account)) && found==0)

{

if(ac.retacno()==n)

{

ac.show\_account();

cout<<"\n\nEnter the new details of account"<<endl;

ac.modify();

int pos=(-1)\*sizeof(account);

File.seekp(pos,ios::cur);

File.write((char \*) &ac, sizeof(account));

cout<<"\n\n\t Record Updated!";

found=1;

}

}

File.close();

if(found==0)

cout<<"\n\n Record Not Found ";

}

// FUNCTION TO DELETE RECORD OF FILE

void delete\_account(int n)

{

account ac;

ifstream inFile;

ofstream outFile;

inFile.open("acc.dat",ios::binary);

if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

return;

}

outFile.open("Temp.dat",ios::binary);

inFile.seekg(0,ios::beg);

while(inFile.read((char \*) &ac, sizeof(account)))

{

if(ac.retacno()!=n)

{

outFile.write((char \*) &ac, sizeof(account));

}

}

inFile.close();

outFile.close();

remove("acc.dat");

rename("Temp.dat","acc.dat");

cout<<"\n\n\tRecord Deleted ..";

}

// FUNCTION TO DISPLAY ALL ACCOUNT DEPOSIT LIST

void display\_all()

{

account ac;

ifstream inFile;

inFile.open("acc.dat",ios::binary);

if(!inFile)

{

cout<<"File could not be open !! Press any Key...";

return;

}

cout<<"\n\n\t\tACCOUNT HOLDER LIST\n\n";

cout<<"====================================================\n";

cout<<"A/c no. NAME Type Balance\n";

cout<<"====================================================\n";

while(inFile.read((char \*) &ac, sizeof(account)))

{

ac.report();

}

inFile.close();

}

// FUNCTION TO DEPOSIT AMOUNTS

void deposit (int n)

{

int amt;

int found=0;

account ac;

fstream File;

File.open("acc.dat", ios::binary|ios::in|ios::out);

if(!File)

{

cout<<"File could not be open !! Press any Key...";

return;

}

while(File.read((char \*) &ac, sizeof(account)) && found==0)

{

if(ac.retacno()==n)

{

ac.show\_account();

{

cout<<"\n\n\tTO DEPOSIT AMOUNT ";

cout<<"\n\nEnter The amount to be deposited : ";

cin>>amt;

ac.dep(amt);

}

int pos=(-1)\* sizeof(ac);

File.seekp(pos,ios::cur);

File.write((char \*) &ac, sizeof(account));

cout<<"\n\n\t Record Updated";

found=1;

}

}

File.close();

if(found==0)

cout<<"\n\n Record Not Found ";

}

// FUNCTION TO WITHDRAW AMOUNTS

void withdraw(int n)

{

int amt;

int found=0;

account ac;

fstream File;

File.open("acc.dat", ios::binary|ios::in|ios::out);

if(!File)

{

cout<<"File could not be open !! Press any Key...";

return;

}

while(File.read((char \*) &ac, sizeof(account)) && found==0)

{

if(ac.retacno()==n)

{

ac.show\_account();

{

cout<<"\n\n\tTO WITHDRAW AMOUNT ";

cout<<"\n\nEnter The amount to be withdrawn : ";

cin>>amt;

int bal=ac.retdeposit()-amt;

if((bal<500 &&ac.rettype()=='S') || (bal<1000 && ac.rettype()=='C'))

cout<<"Insufficient balance";

else

ac.draw(amt);

}

int pos=(-1)\* sizeof(ac);

File.seekp(pos,ios::cur);

File.write((char \*) &ac, sizeof(account));

cout<<"\n\n\t Record Updated";

found=1;

}

}

File.close();

if(found==0)

cout<<"\n\n Record Not Found ";

}

// INTRODUCTION FUNCTION

void intro()

{

int pas;

cout<<"\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<endl;

cout<<"\t\t BANK MANAGEMENT SYSTEM ";

cout<<endl;

cout<<"\t\t COMPLETE BANKING SOLUTIONS!!! ";

cout<<endl;

cout<<"\t\t \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*";

cout<<endl;

cout<<endl;

cout<<endl;

for(int i=0;i<3;i++)

{

cout<<"\nEnter the password : ";

cin>>pas;

if(pas==1234)

break;

if(i==2)

{

cout<<"\nWrong password ";

cout<<"\nExiting ";

getch();

exit(0);

}

else

if(pas!=1234)

{

cout<<"\nWrong password ";

cout<<"\nEnter again";

}

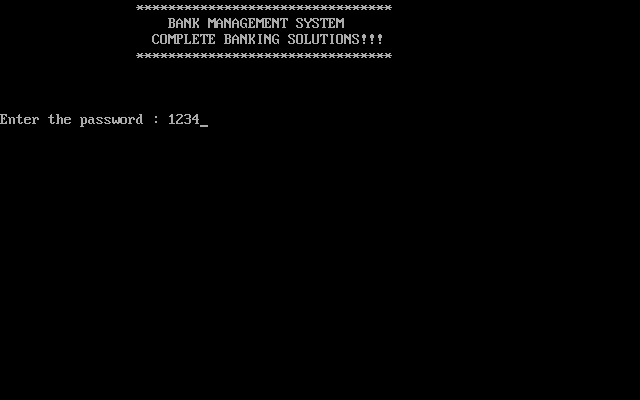
}

getch();

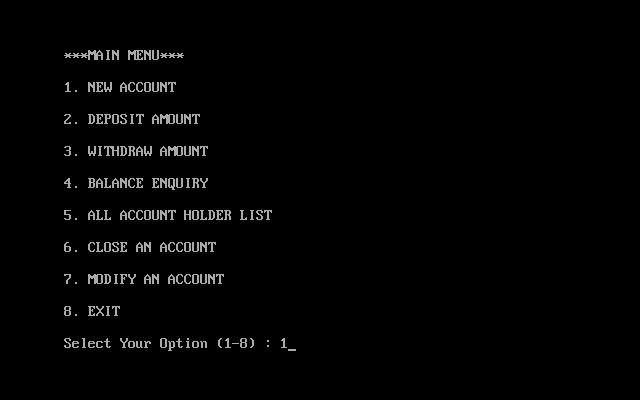
}

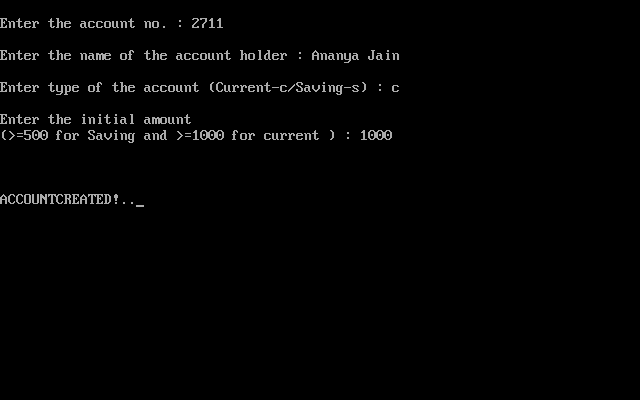
**OUTPUTS**

OPENING SCREEN :-

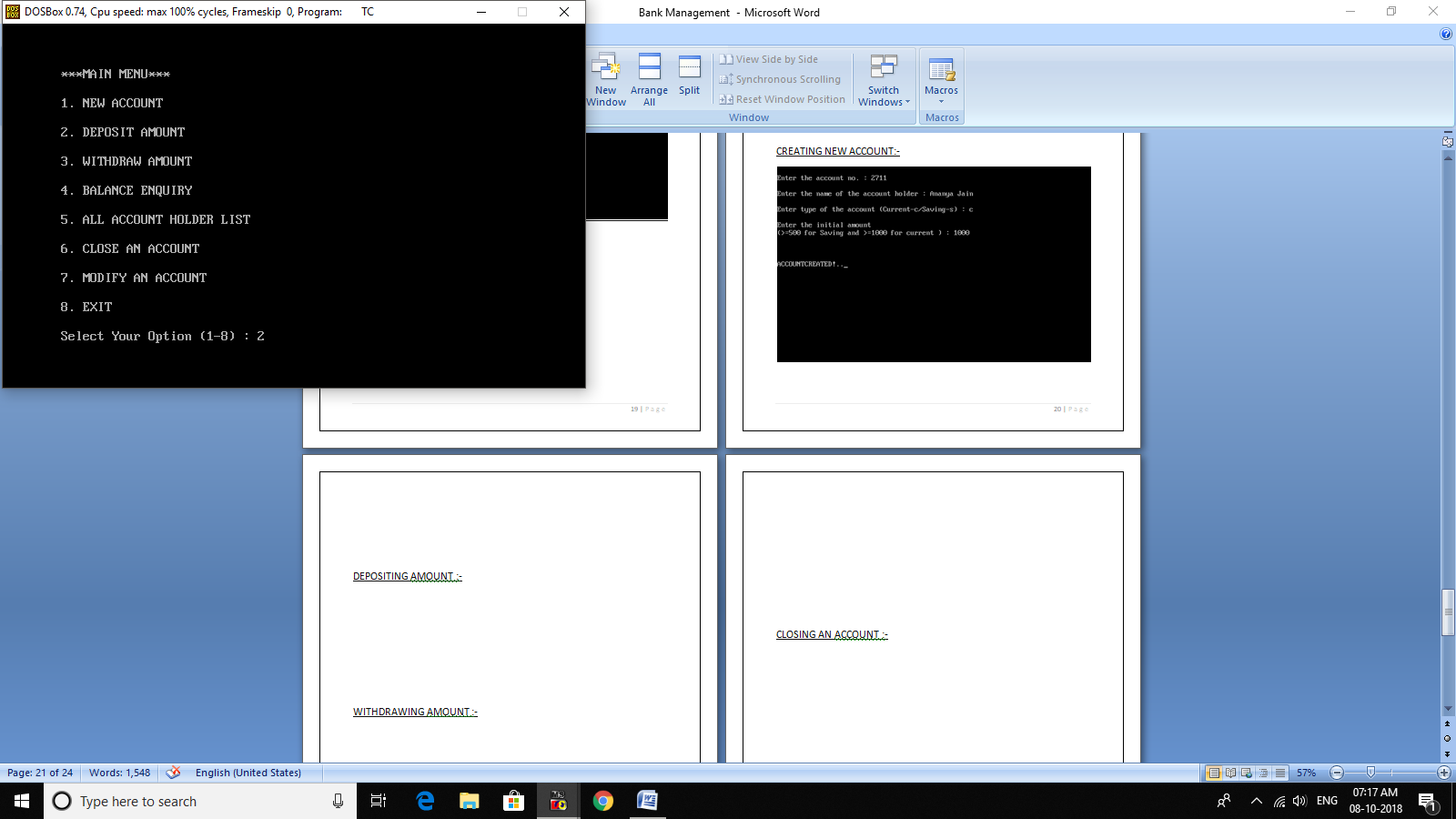


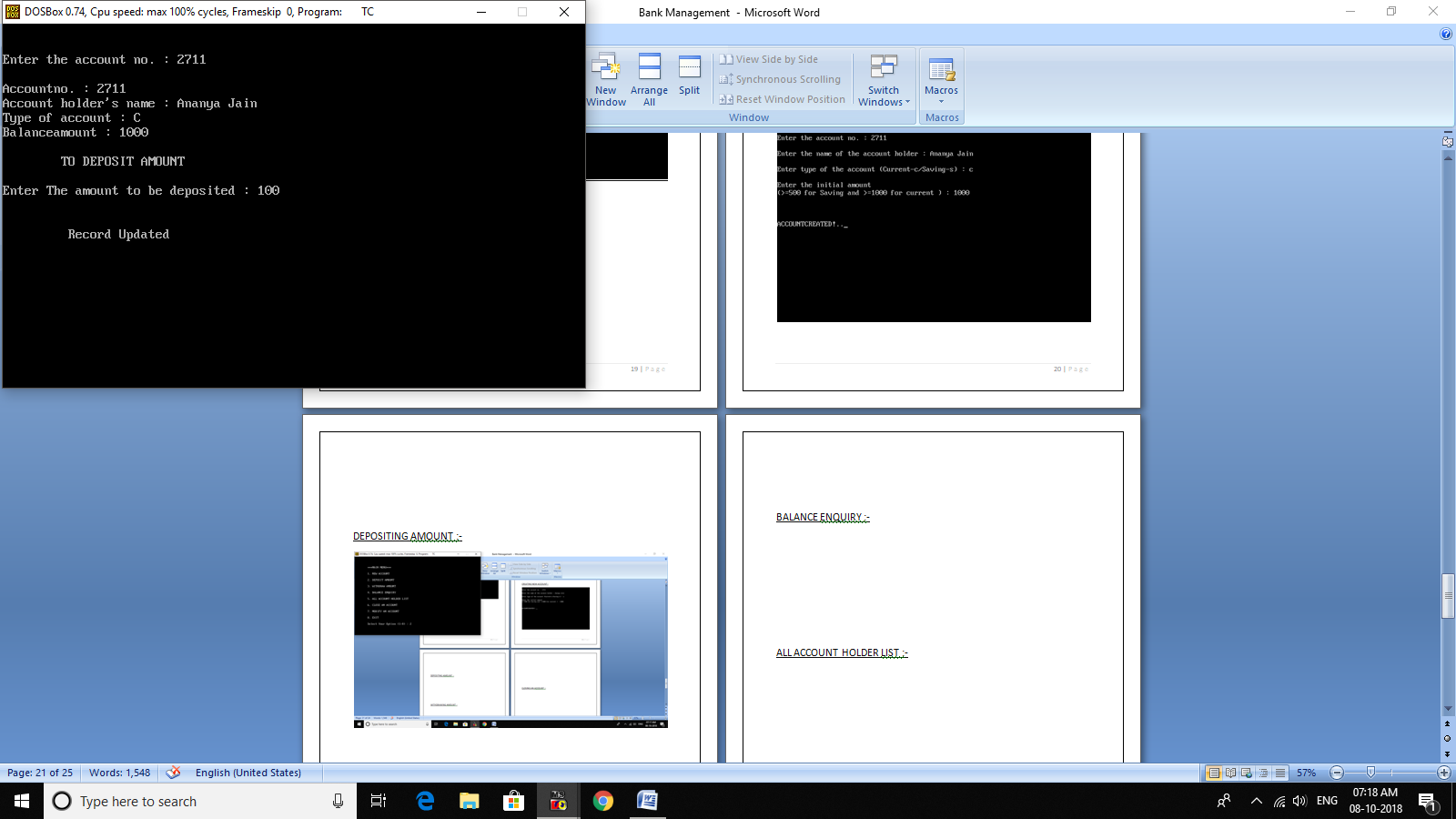
CREATING NEW ACCOUNT:-



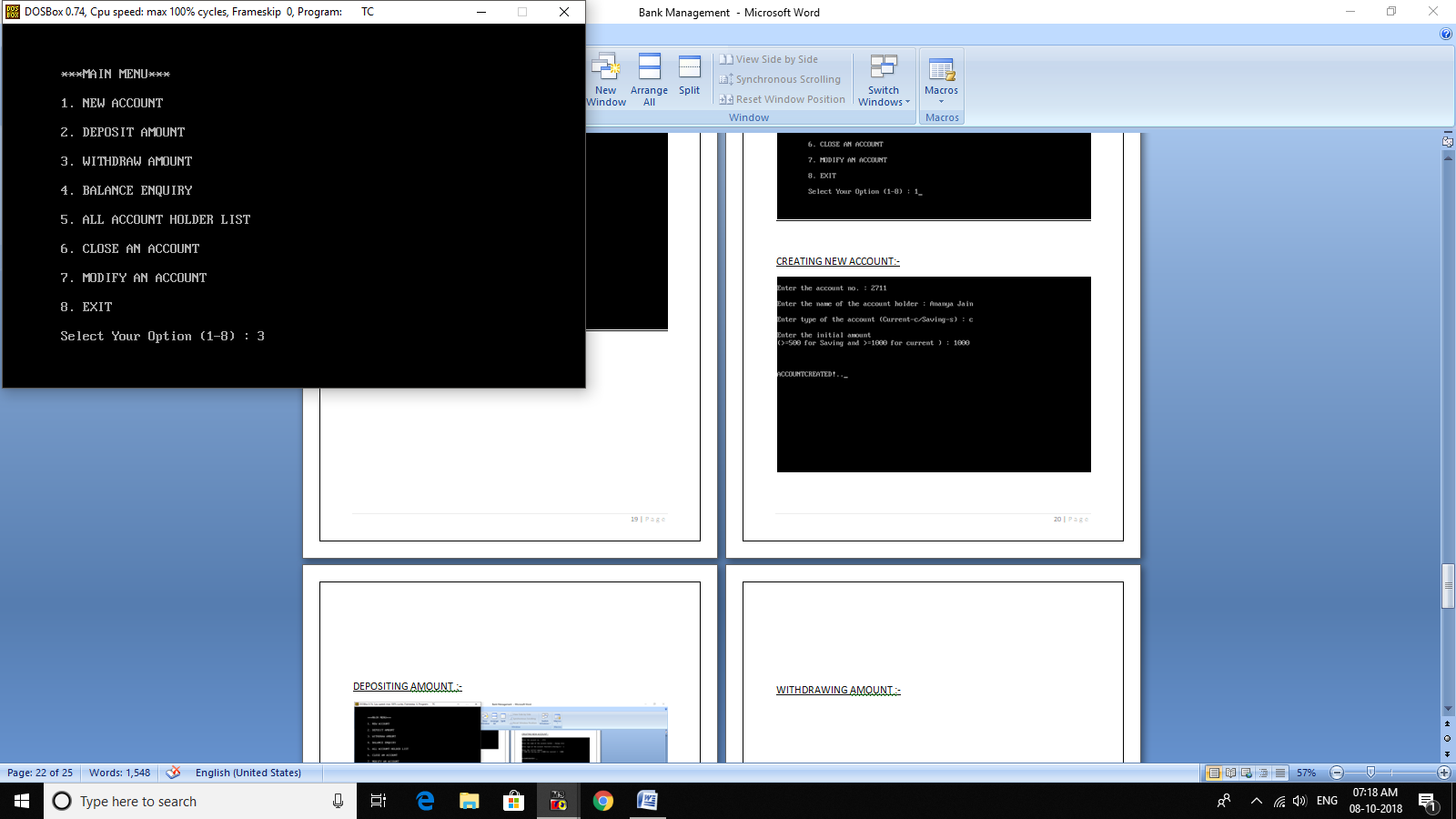


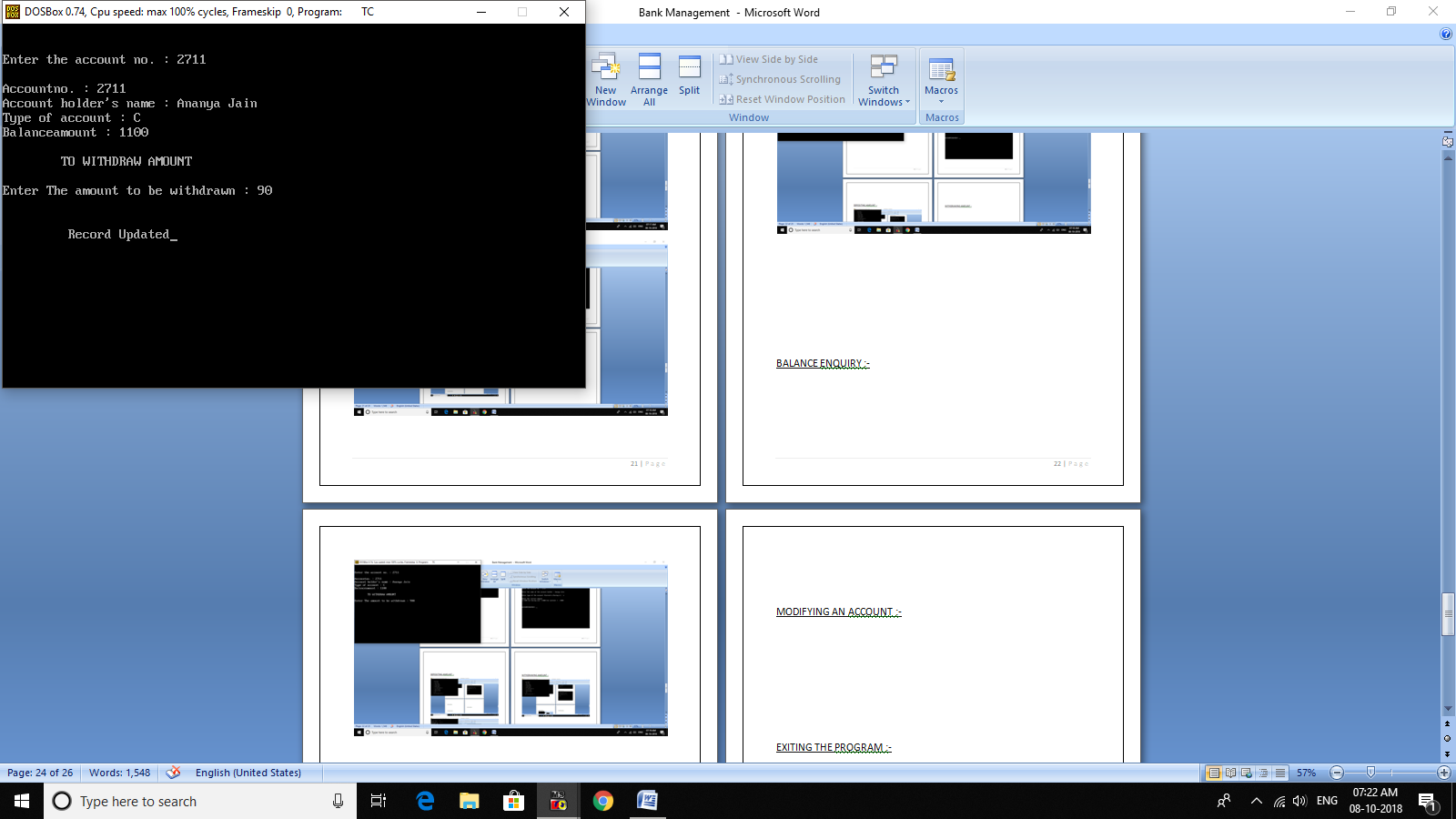
DEPOSITING AMOUNT :-



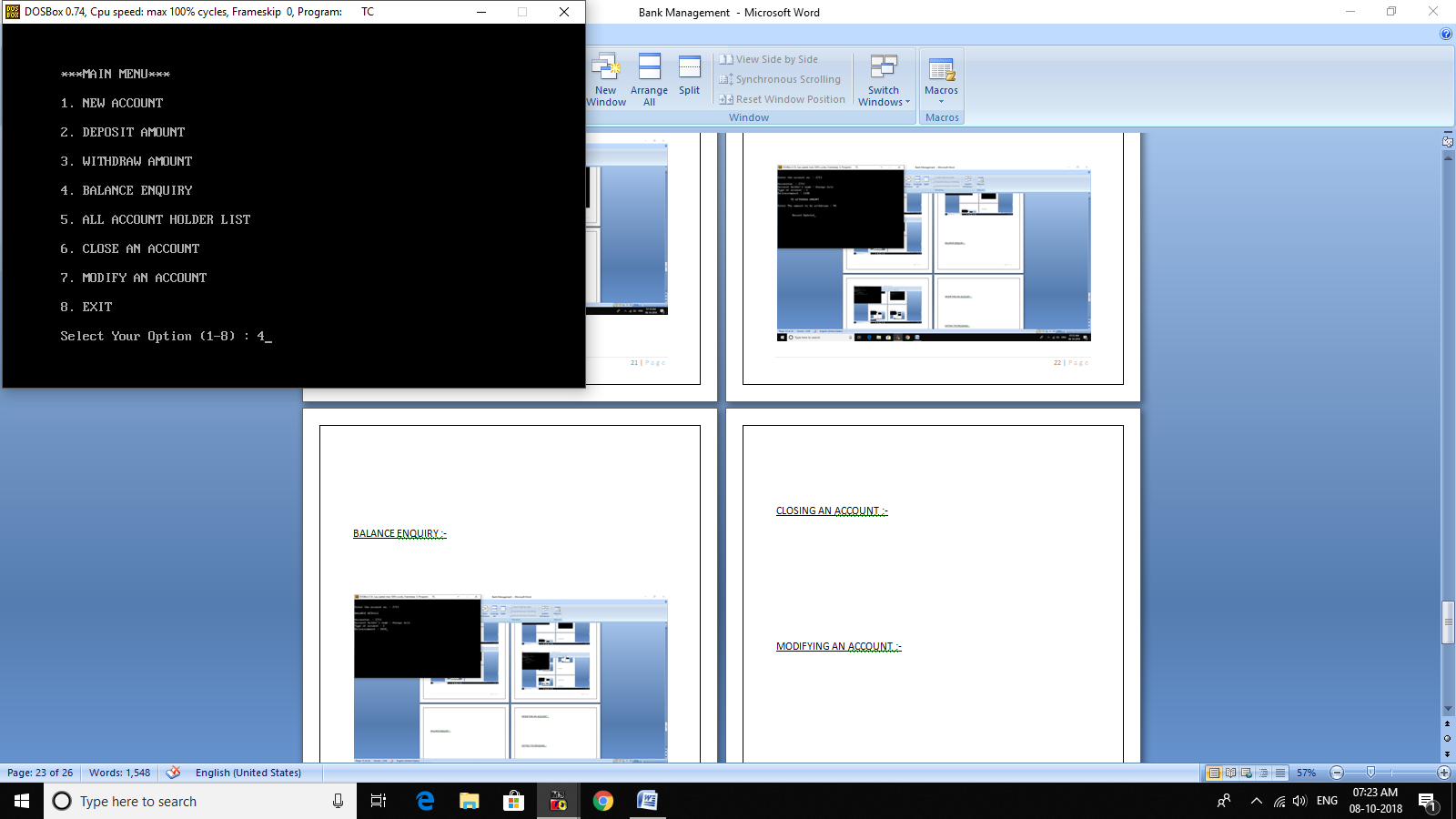


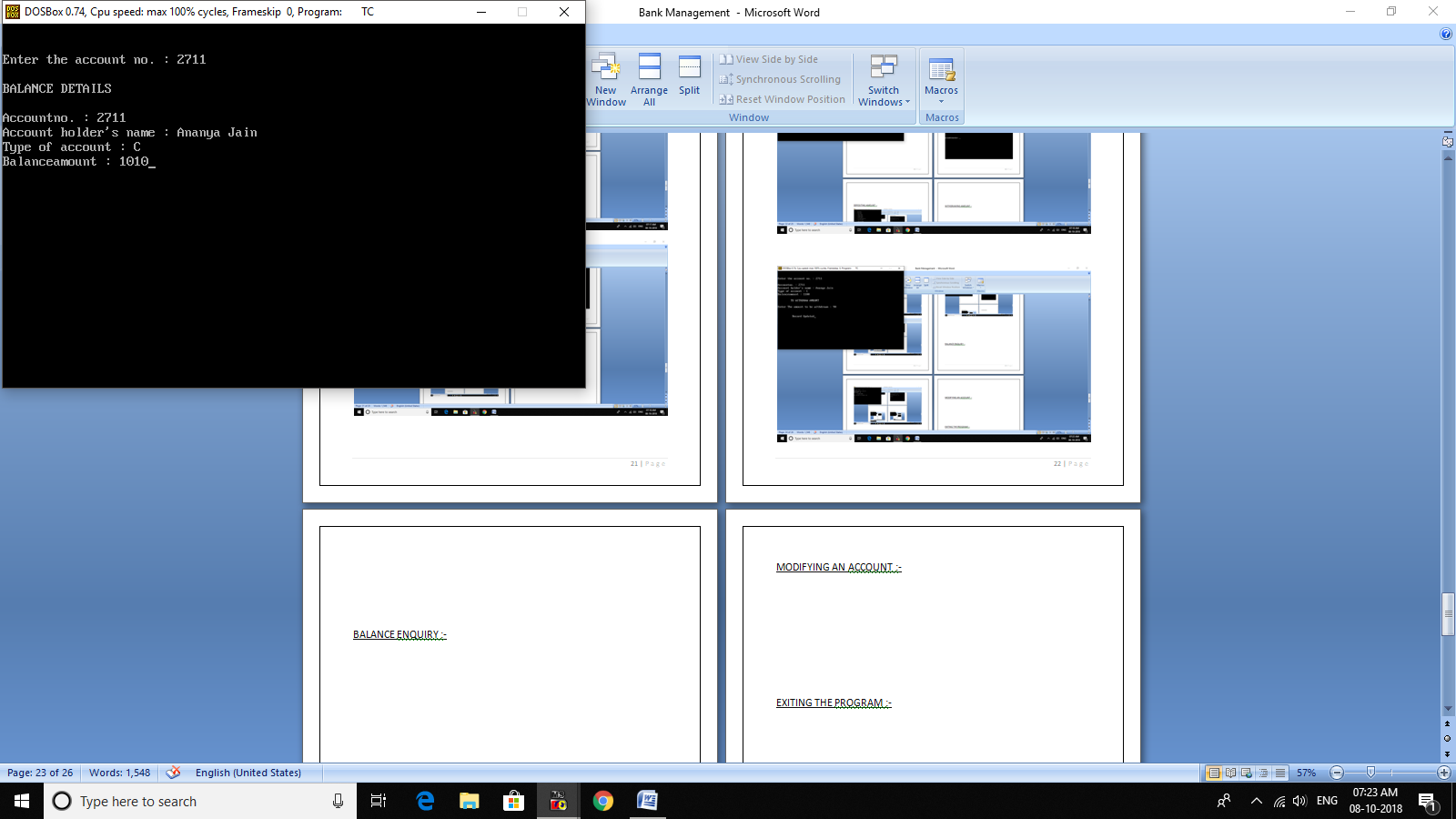
WITHDRAWING AMOUNT :-



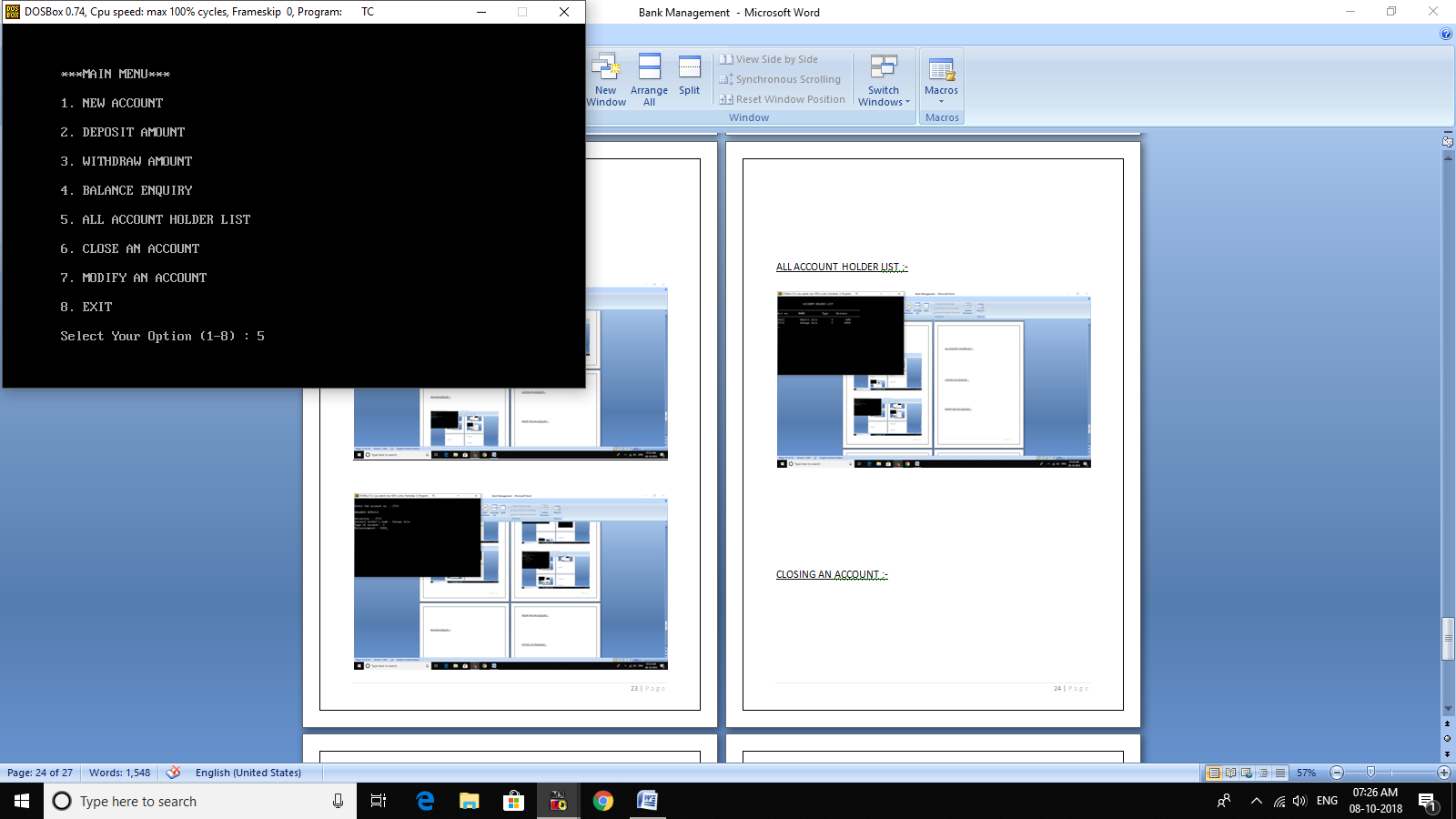


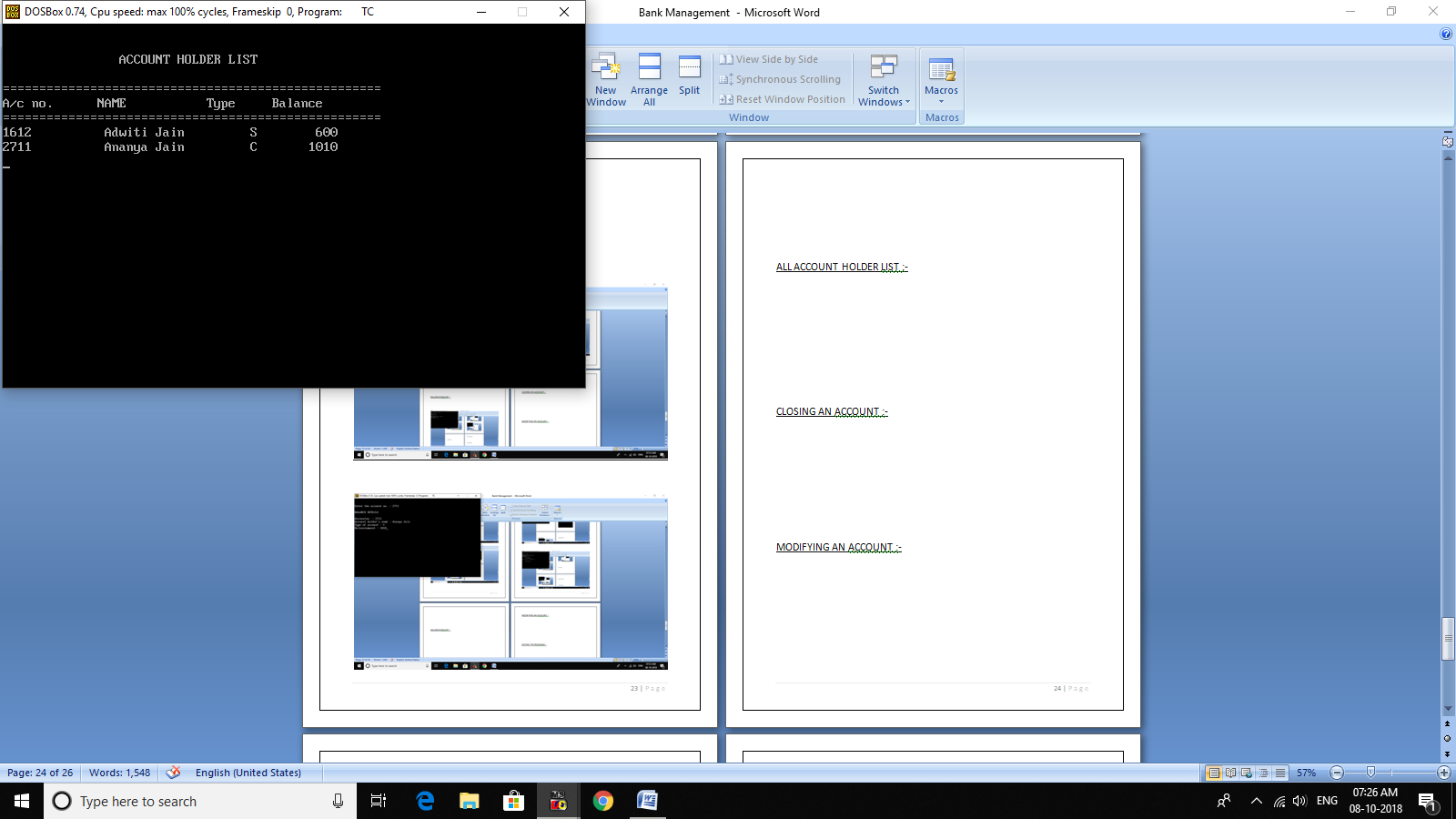
BALANCE ENQUIRY :-



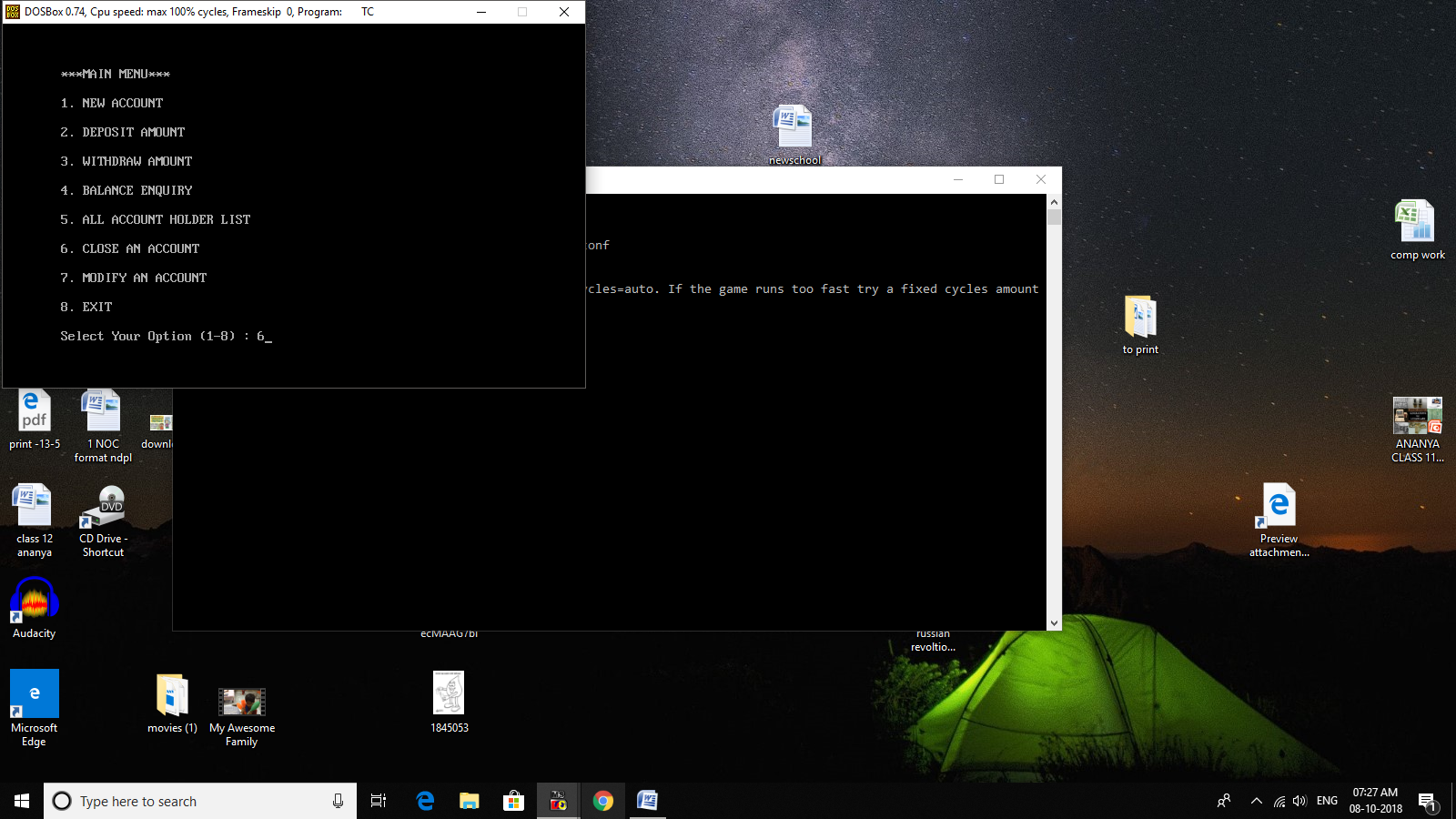


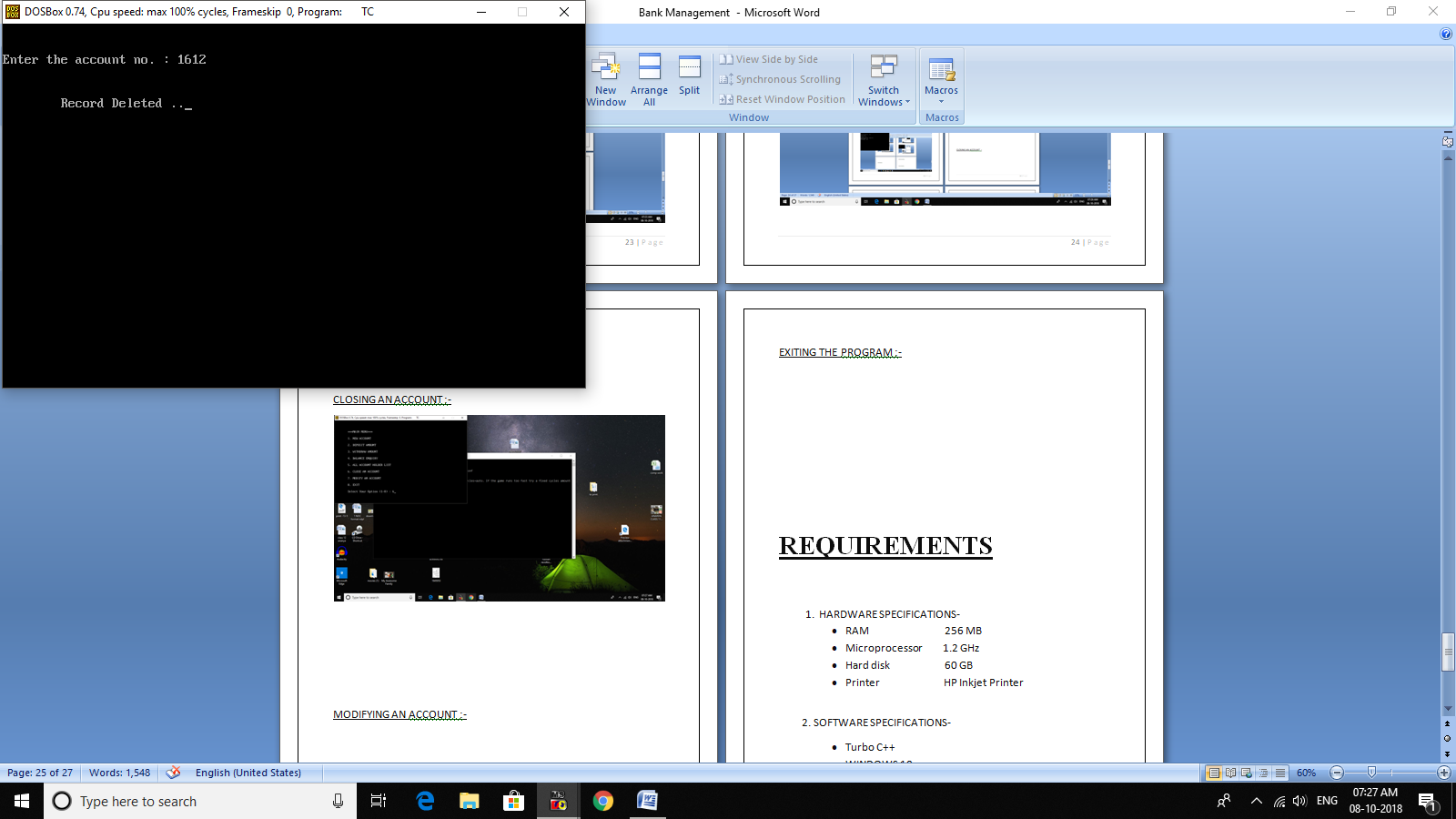
ALL ACCOUNT HOLDER LIST :-



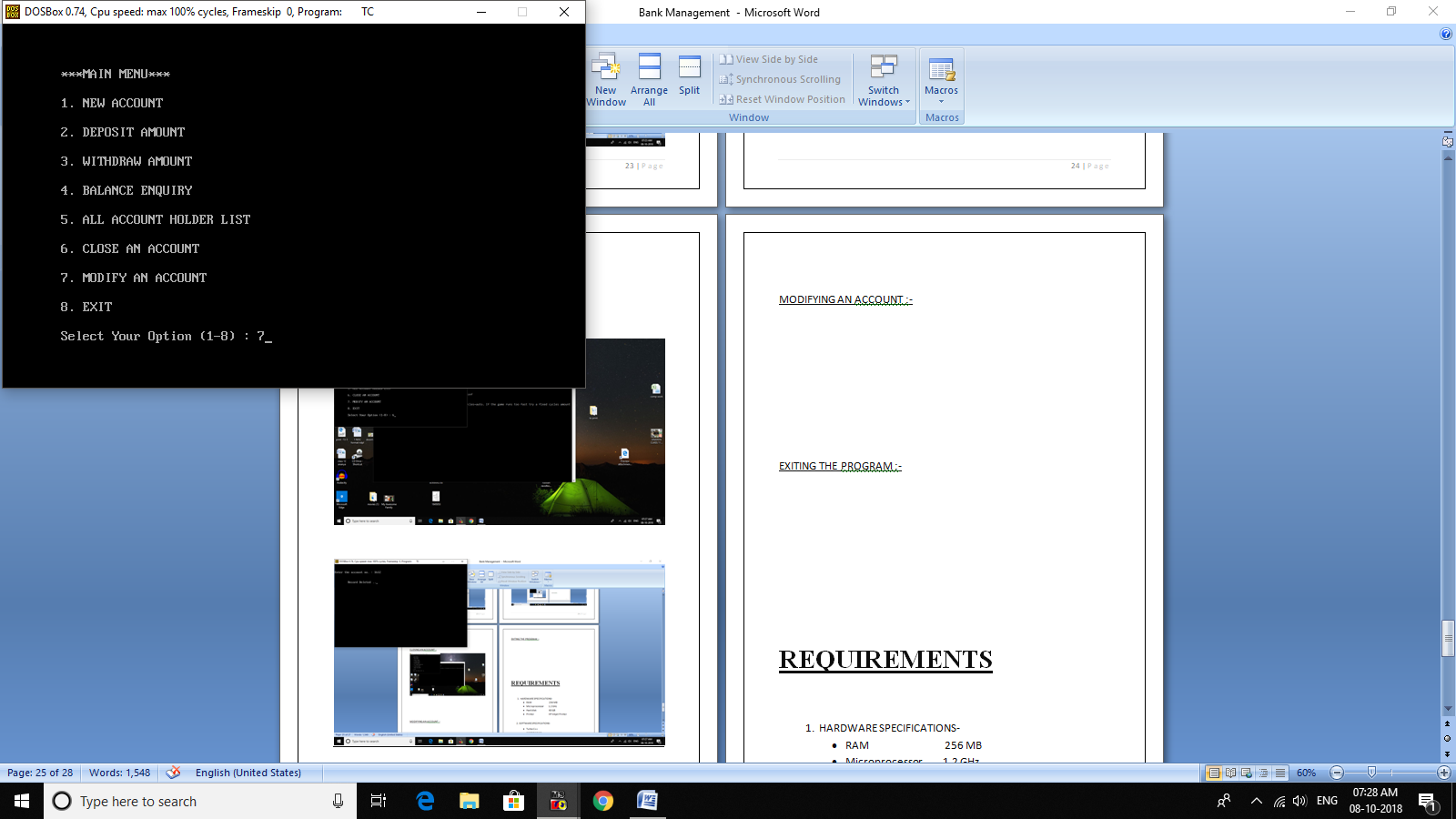


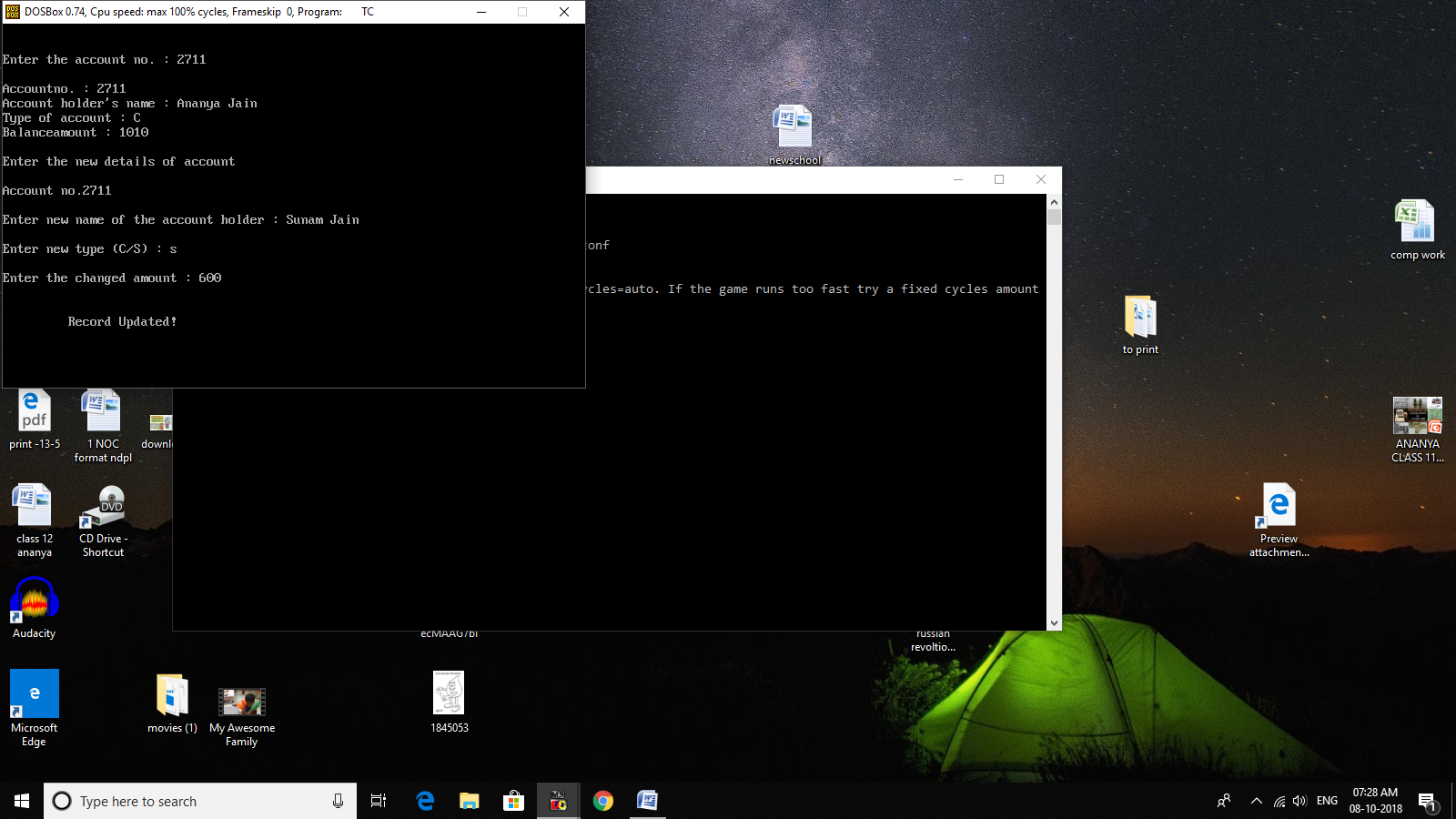
CLOSING AN ACCOUNT :-



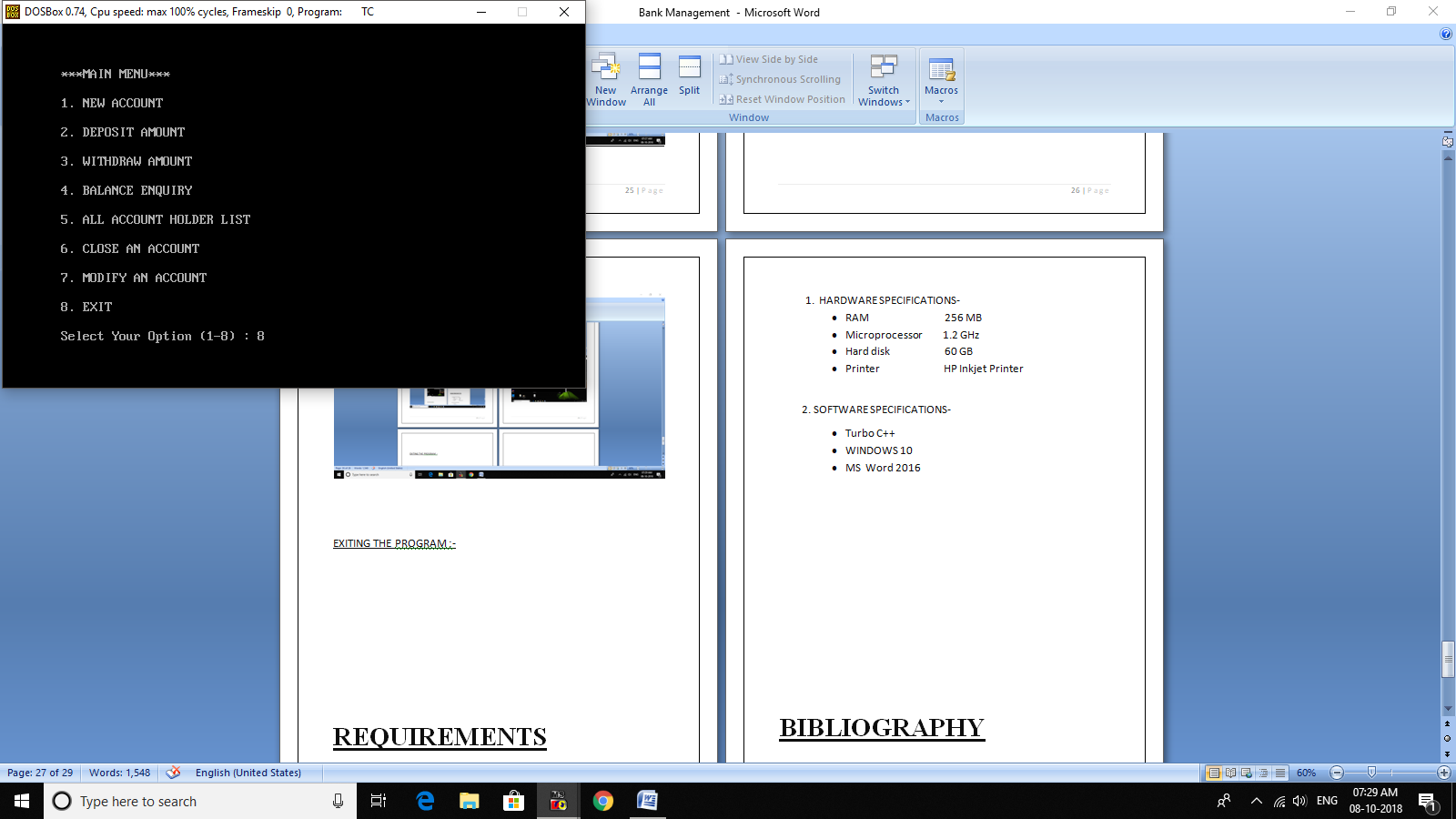


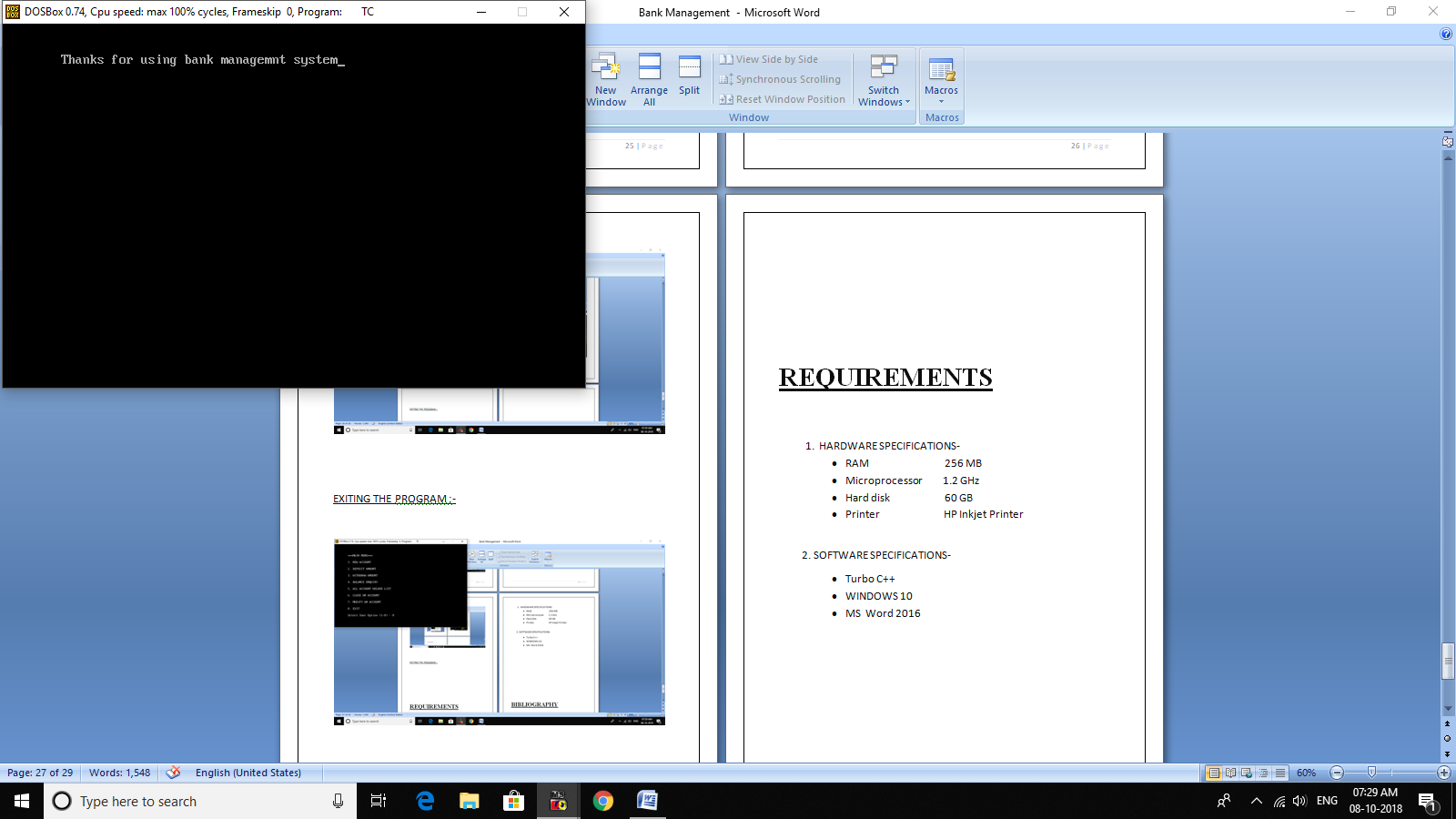
MODIFYING AN ACCOUNT :-





EXITING THE PROGRAM :-





**REQUIREMENTS**

1. HARDWARE SPECIFICATIONS-

* RAM 256 MB
* Microprocessor 1.2 GHz
* Hard disk 60 GB
* Printer HP Inkjet Printer

2. SOFTWARE SPECIFICATIONS-

* Turbo C++
* WINDOWS 10
* MS Word 2016

**WAYS TO IMPROVE**

1. This software can be frther improved by…….

**BIBLIOGRAPHY**

REFERENCES-

1. Books:

* An Overview of C++ Programming Language by Bjarne Stroustrup
* Computer Science C++ by Sumita Arora

1. Websites:

* http://www.programmersheaven.com
* http://stackoverflow.com
* http://www.dreamincode.net