**Banking Management**



Ayushi Amin

12A

**DELHI PRIVATE SCHOOL**

**SHARJAH**



DEPARTMENT OF COMPUTER SCIENCE

**CERTIFICATE**

Certified that the work in this file is the bonafide work of \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ of class XII \_\_\_\_\_ Roll Number \_\_\_\_\_\_\_\_\_\_\_\_\_\_ recorded in the school labs during the academic year 2018 – 2019.

Date :\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Teacher in charge External Examiner

**Index**

|  |  |  |
| --- | --- | --- |
| **S.No** | **Title** | **Page No.** |
| 1 | Acknowledgement | 4 |
| 2 | Description   * Data required for input * Output given * System requirements | 5-6 |
| 3 | Source Code | 7-17 |
| 4 | Screenshots | 18-20 |
| 5 | Bibliography | 21 |

**Acknowledgement**

Primarily I would like to thank God for giving me the strength to be being able to complete this project with success.

I would like to extend my gratitude to the Principle and Director, Mrs. Vandana Marwaha for providing me with all the facility that was required.

I would also like to express my special thanks of gratitude to my teacher, Mrs. Daliya, who gave me the golden opportunity to do this wonderful project of Computer Science and for her able guidance and support in completing my project.

Secondly, I would like to thank my parents who helped me a lot in finalizing this project within the limited time frame.

**Description**

The theme of this program is handling bank accounts.

The program first starts with the class account where many essential functions are performed.

During execution, the user is first asked to create a new account and this information is stored in the file – account.dat

The user can deposit and withdraw cash from his account and also he can view the details of his account after any updating.

If the user feels like closing his account, he has the opportunity to do so.

Concepts of Object Oriented Programming like data abstraction, data encapsulation, modularity and inheritance have been used in the program to give it more object based approach rather than a procedural approach and also develop real world relationships.

Functions are used to break the program into several segments.

Accessor functions have been used to access the data but not modify it.

Mutator functions have been used to modify the data.

**Data Required for Input**

1. name Account Holder’s Name
2. acc\_no Account Number
3. acc\_type Type of account(Current/Savings)
4. deposit

**Functions Performed**

* Create account
* Deposit amount
* Withdraw amount
* Check balance
* Impose penalty (current)
* Calculate interest (savings)
* Exit

**System Requirements**

* Minimum of 128 MB RAM
* Minimum processor type – Pentium II
* Minimum processor speed – 400 MHz
* Additional Requirements – CD-ROM, HDMI monitor, mouse and keyboard, also other compatible devices.
* Operating System – Microsoft Windows 98 or better, linux ubunto or better.
* C++ Compiler – Borland c++, Turbo c++, Visual Studio 2015 or better.

**Source Code**

#include<iostream.h>

#include<stdio.h>

#include<conio.h>

#include<fstream.h>

#include<process.h>

class account

{

char name[20];

int acc\_no;

char acc\_type;

int pin;

int deposit;

public:

void get\_accinfo();

void menu();

void display\_accinfo();

void depos(int);

void draw(int);

int retacc\_no();

int retpin();

int retdeposit();

char retacc\_type();

};

void account::get\_accinfo()

{

cout<<"\n\nEnter Customer Name : ";

gets(name);

cout<<"\nEnter Account Number : ";

cin>>acc\_no;

cout<<"\nEnter Account Type(Enter C for current or S for savings) : ";

cin>>acc\_type;

cout<<"\nEnter initial ammount(>=500 for saving and >=1000 for current) : ";

cin>>deposit;

cout<<"\nEnter pin number : ";

cin>>pin;

cout<<"\n\nAccount succesfully created";

}

void account::display\_accinfo()

{

cout<<"\n\nCustomer Name :- "<<name;

cout<<"\nAccount Number :- "<<acc\_no;

cout<<"\nAccount Type :- "<<acc\_type;

cout<<"\nBalance amount"<<deposit;

}

void account::depos(int x)

{

deposit+=x;

}

void account::draw(int x)

{

deposit-=x;

}

int account::retacc\_no()

{

return acc\_no;

}

int account::retdeposit()

{

return deposit;

}

char account::retacc\_type()

{

return acc\_type;

}

int account::retpin()

{

return pin;

}

int c=0;

void write()

{

account a;

ofstream o1;

o1.open("account.dat",ios::binary|ios::app);

a.get\_accinfo();

o1.write((char\*)&a,sizeof(account));

o1.close();

}

void login()

{

account a;

ifstream i1("account.dat",ios::binary);

if(!i1)

{

cout<<"error...File not found";

return;

}

cout<<endl;

int lid,pinno;

cout<<"Account no : "<<endl;

cin>>lid;

cout<<endl;

cout<<"PIN : "<<endl;

cin>>pinno;

cout<<endl;

int i=0;

while(i1.read((char\*)&a,sizeof(a)))

{

if((a.retacc\_no()==lid)&&(a.retpin()==pinno))

{

i++;

a.menu();

}

}

if(i==0)

cout<<"invalid\n";

i1.close();

}

void disp\_acc(int n)

{

int flag=0;

account a;

ifstream i1;

i1.open("account.dat",ios::binary);

if(!i1)

{

cout<<"file could not be opened"<<endl;

return;

}

cout<<"\nBalnce Details\n";

while(i1.read((char\*)&a,sizeof(account)))

{

if(a.retacc\_no()==n)

{

a.display\_accinfo();

flag=1;

}

}

i1.close();

if(flag==0)

cout<<"\nAccount doesn't exist";

}

void del\_acc(int n)

{

account a;

ifstream i1;

ofstream o1;

i1.open("account.dat",ios::binary);

if(!i1)

{

cout<<"file could not be opened"<<endl;

return;

}

o1.open("new.dat",ios::binary);

i1.seekg(0,ios::beg);

while(i1.read((char\*)&a,sizeof(a)))

{

if(a.retacc\_no()!=n)

{

o1.write((char\*)&a,sizeof(a));

}

}

i1.close();

o1.close();

remove("account.dat");

rename("new.dat","account.dat");

cout<<"\n\n\tRecord deleted";

c=0;

return;

}

void deposit\_withdraw(int n,int opt)

{

int amt,flag=0;

account a;

fstream f;

f.open("account.dat",ios::binary|ios::in|ios::out);

if(!f)

{

cout<<"file could not be opened"<<endl;

return;

}

while(!f.eof() && flag==0)

{

f.read((char\*)&a,sizeof(a));

if(a.retacc\_no()==n)

{

a.display\_accinfo();

if(opt==1)

{

cout<<"\n\n\tDEPOSIT AMT";

cout<<"\n\nEnter amt to be deposited";

cin>>amt;

a.depos(amt);

}

else if(opt==2)

{

cout<<"\n\n\tWITHDRAW AMT";

cout<<"\n\nEnter amt to be withdrawn";

cin>>amt;

int bal=a.retdeposit()-amt;

if((bal<500&&a.retacc\_type()=='S')||(bal<1000&&a.retacc\_type()=='C'))

cout<<"insufficient balnce"<<endl;

else

a.draw(amt);

}

f.seekp((-1)\*sizeof(a),ios::cur);

f.write((char\*)&a,sizeof(a));

cout<<"\n\n\tRecord updated";

flag=1;

}

}

f.close();

if(flag==0)

cout<<"\n\n Record not found";

}

void account::menu()

{

int ch;

do

{

cout<<"WELCOME "<<name<<endl;

cout<<"\nChoose Your Choice\n";

cout<<"1)Deposit\n";

cout<<"2)Withdraw\n";

cout<<"3)Balance enquiry\n";

cout<<"4)Close account\n";

cout<<"5)Log out\n";

cout<<"Choose Your choice";

cin>>ch;

switch(ch)

{

case 1: deposit\_withdraw(acc\_no,1);

break;

case 2: deposit\_withdraw(acc\_no,2);

break;

case 3: disp\_acc(acc\_no);

break;

case 4: c=acc\_no;

break;

case 5: cout<<"Thnak you, you have successfully logged out"<<endl;

break;

}

}

while(ch>=1&&ch<=5);

}

void main()

{

cout<<" HSBC BANK "<<endl;

int choice;

do

{

if(c!=0)

del\_acc(c);

cout<<" Welcome User1 What would you like to do? "<<endl;

cout<<"1)Create account"<<endl;

cout<<"2)login into account"<<endl;

cout<<"3)exit"<<endl;

cout<<"enter choice"<<endl;

cin>>choice;

switch(choice)

{

case 1: write();

break;

case 2: login();

break;

case 3: exit(0);

default: cout<<"enter correct choice";

}

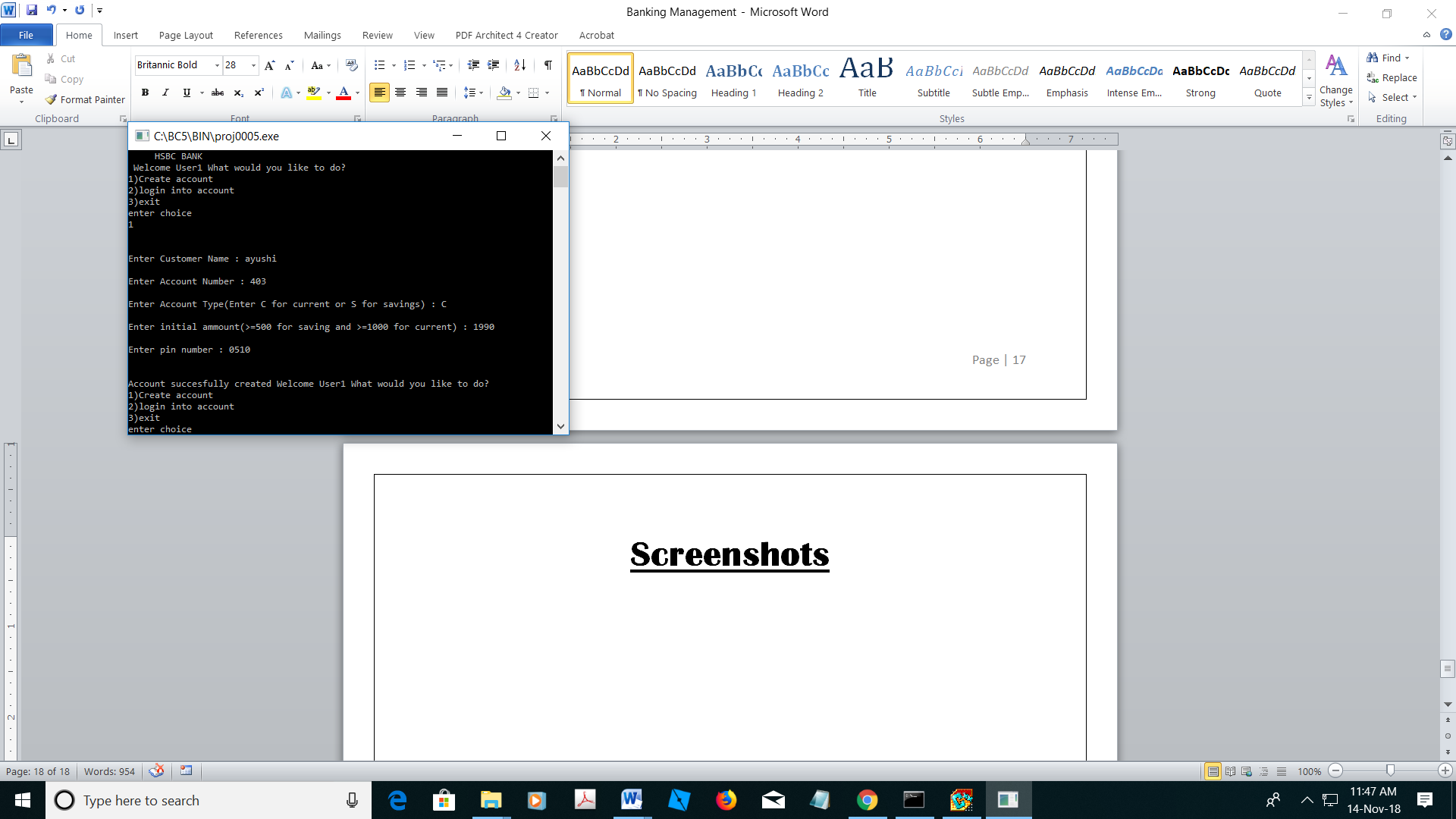
}

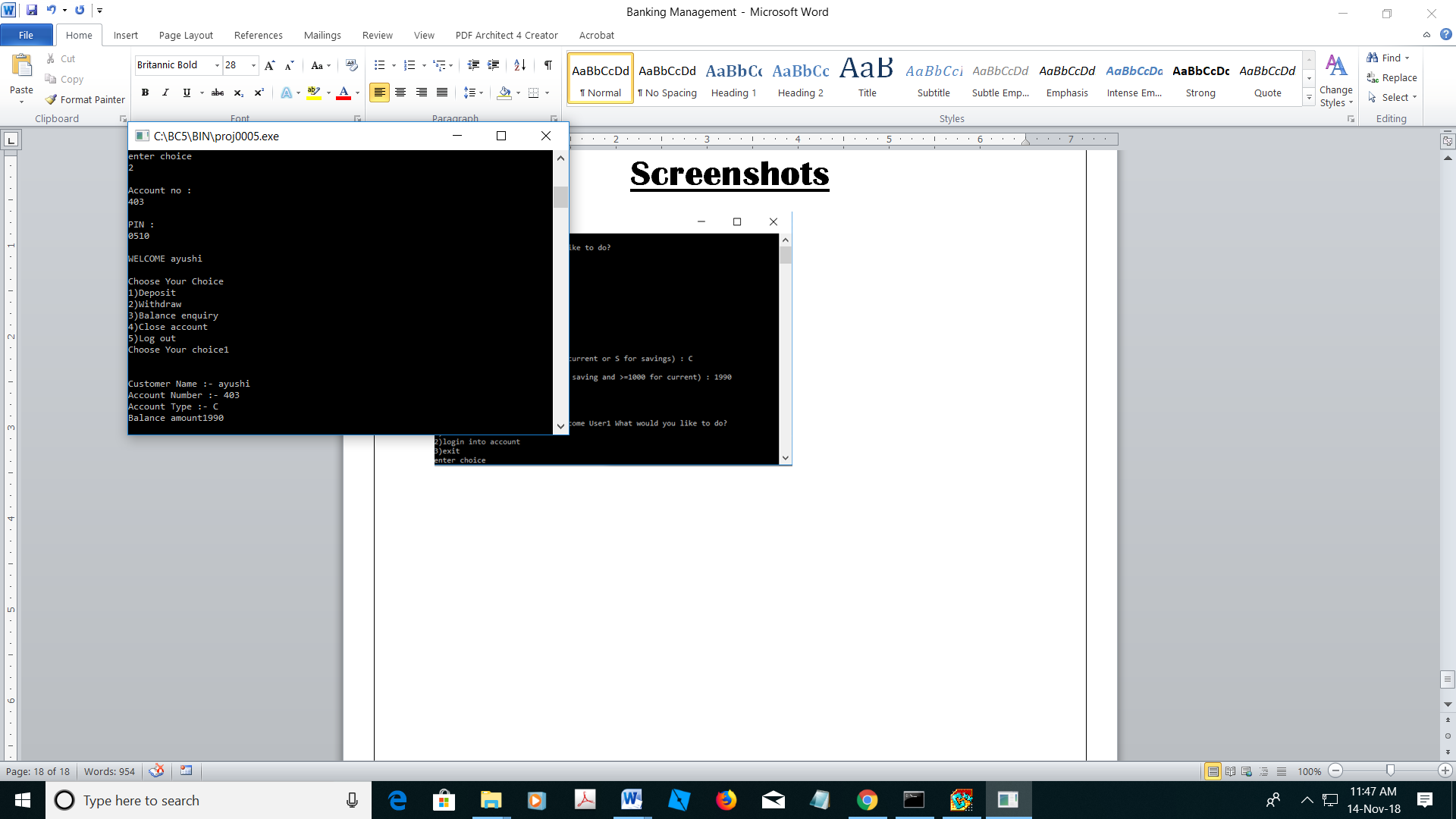
while(choice>=1&&choice<=3);

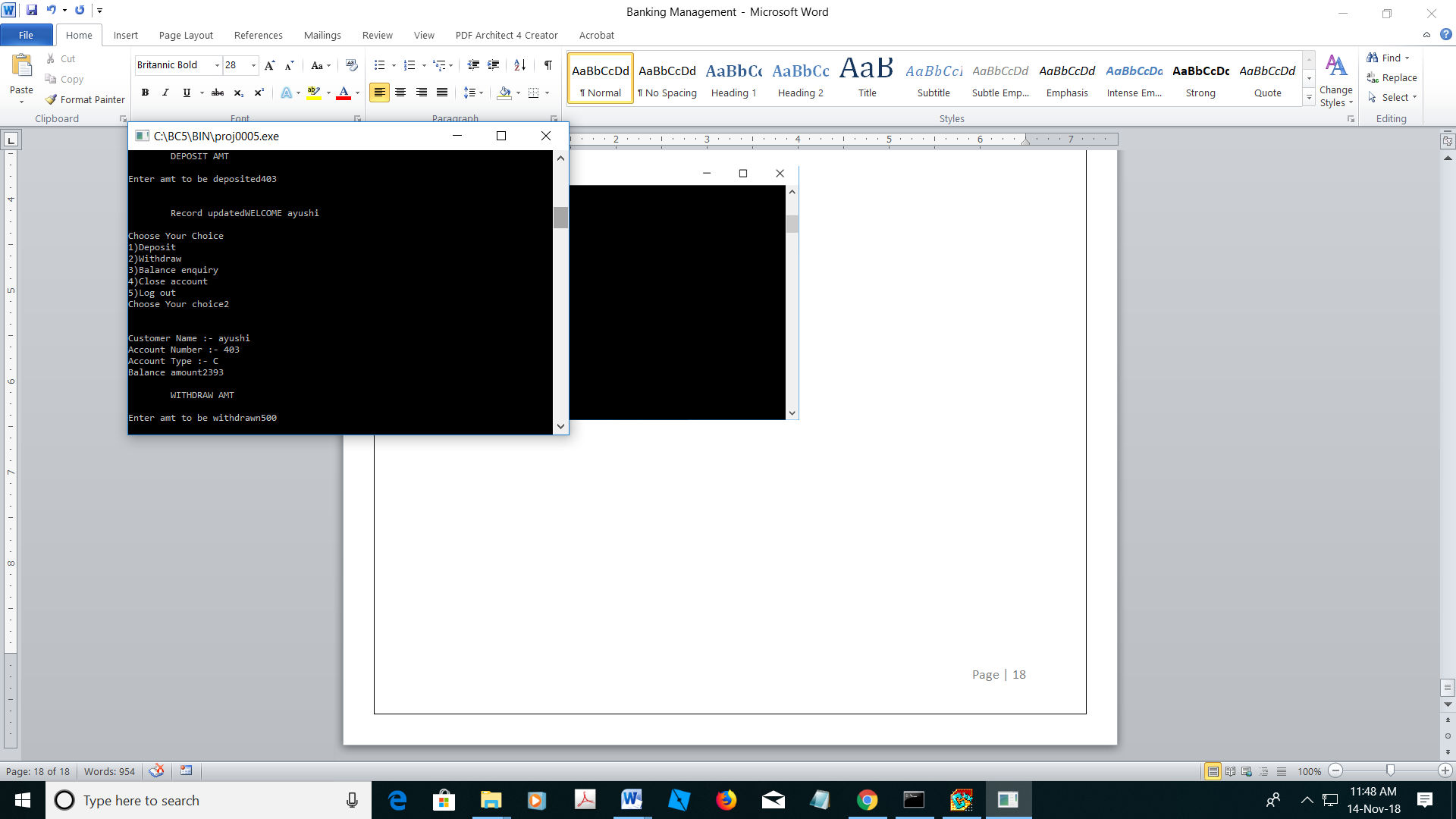
getch();

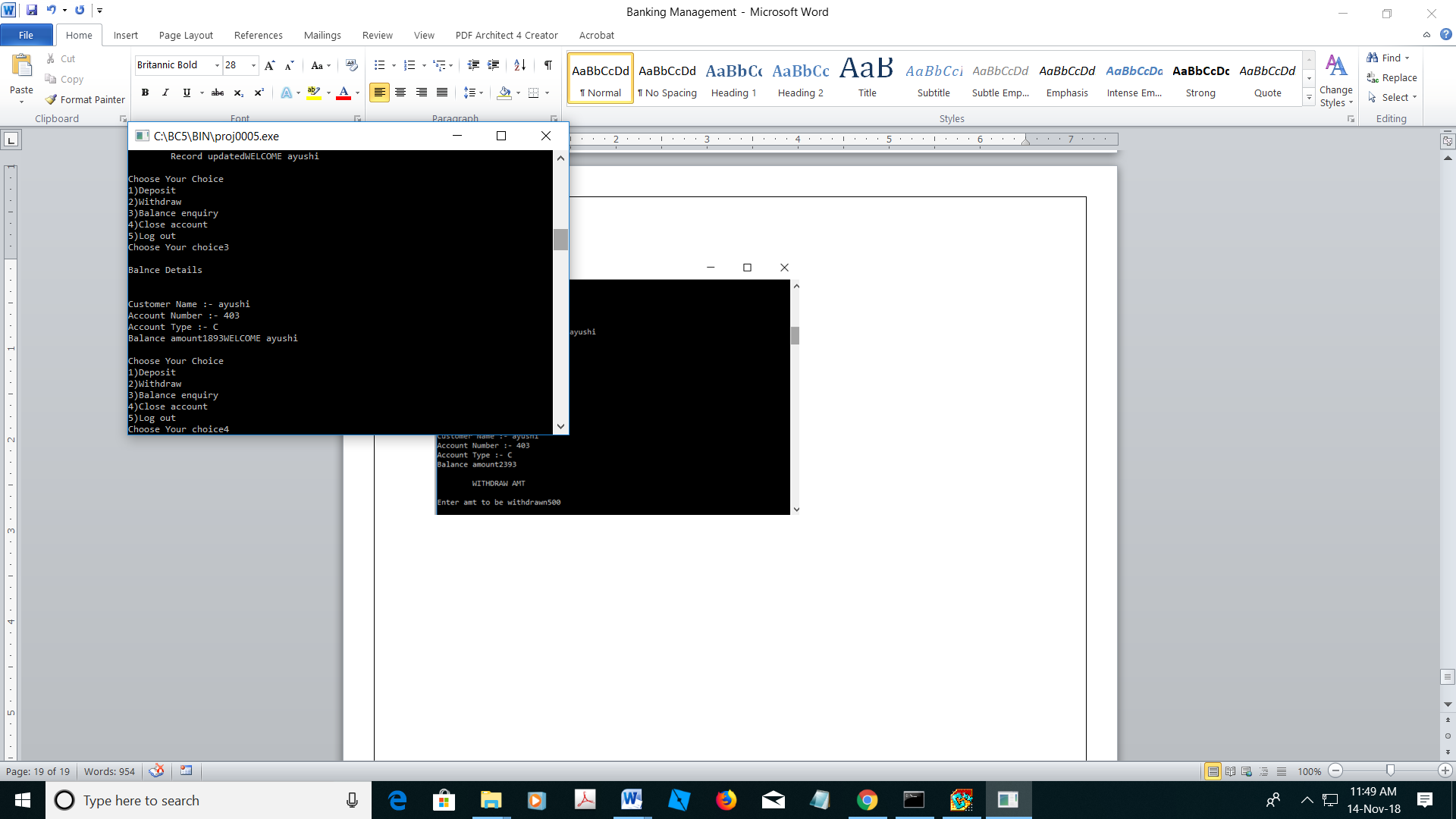
}

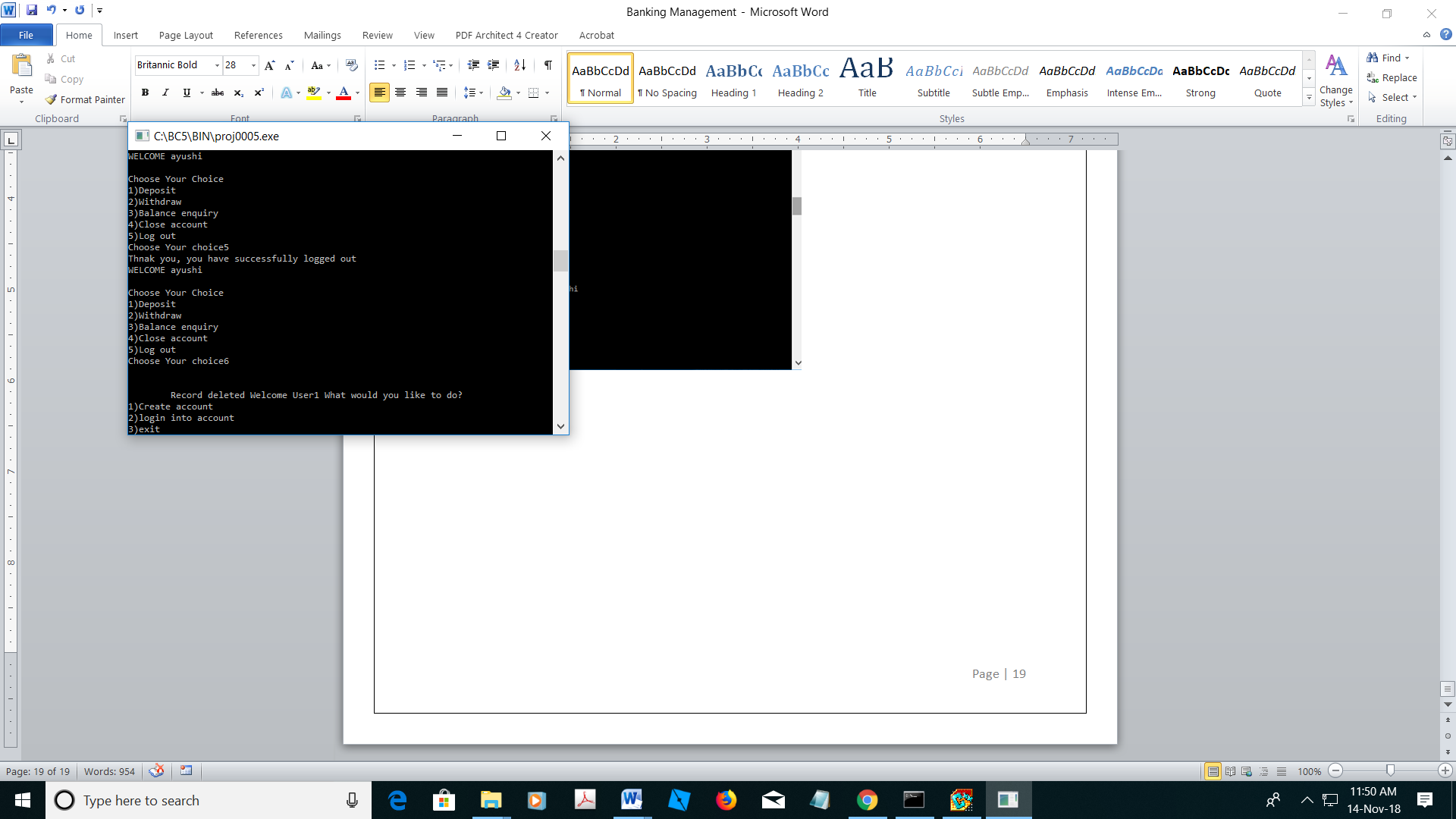
**Screenshots**

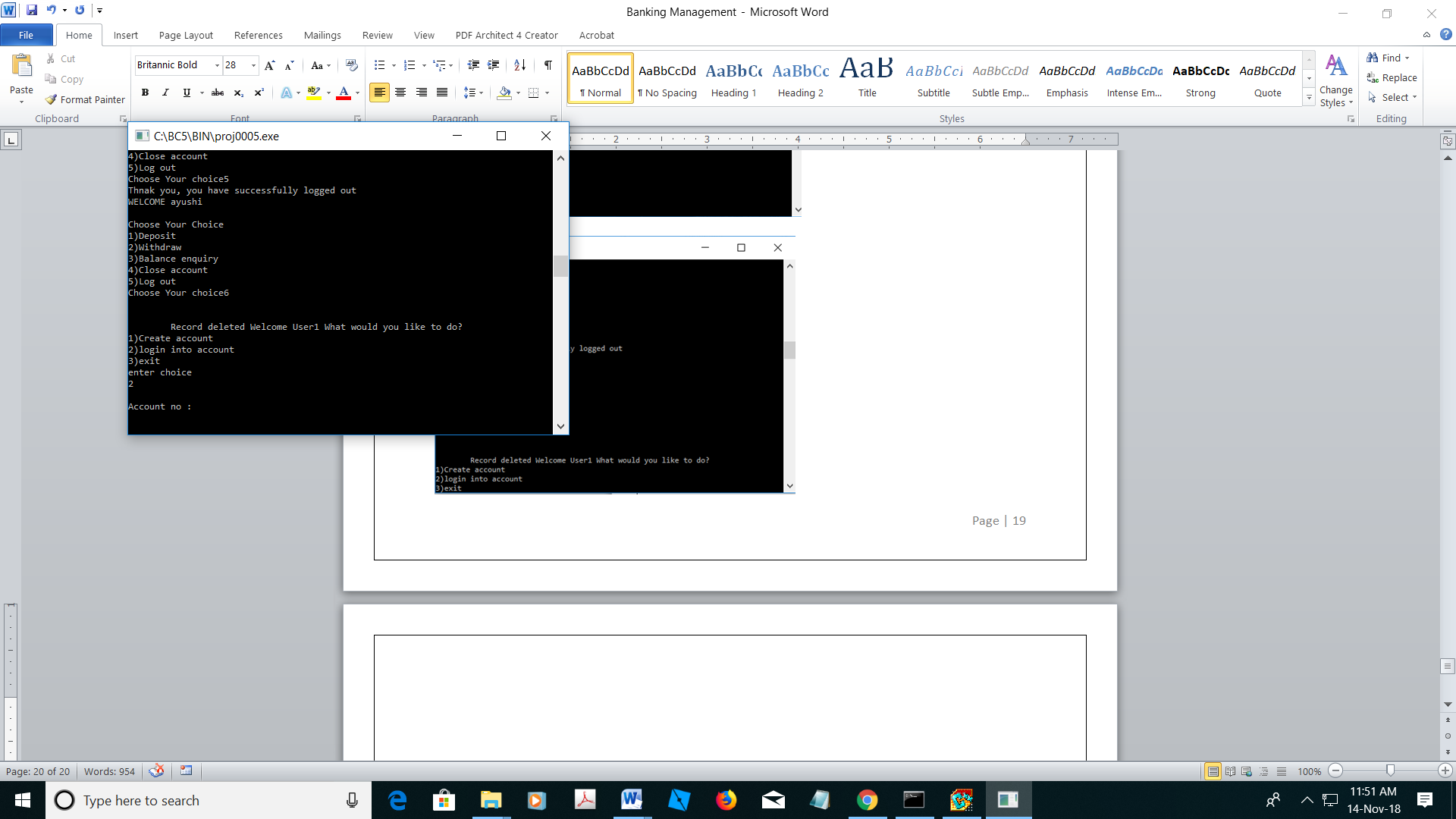












**Bibliography**

1. Computer science with C++ textbook
2. [www.stackoverflow.com](http://www.stackoverflow.com)
3. www.borland.public.cpp.borlandcpp.narkive.com