Bradley Turner

ESE224

12/4/2015

**ESE224 Final Project Report**

**Objective:**

The purpose behind the final project was to expose students to miniature application development in the form of a banking system. It also brought together everything in the ESE 224 curriculum so students can really demonstrate their newfound knowledge and specific coding styles. The program is to function almost as if it were an ATM terminal for a small bank. There are two user classes the client class and the manager class. The client class functions almost exactly like an ATM allowing the user to view their account balance, and withdraw and deposit funds. The manager class acts as an administrative class allowing the user to create, delete, or modify client accounts much like a bank teller would.

**Specifications of this Project:**

The two user classes described above consist of one manager and five users. The bonus exercises were also complete showing asterisks (\*) for password inputs and allowing executive money transfers in the manager class. The accepted user names for the manager and client classes are displayed below:

Manager:

* Username: manager
* Password: password

Clients: A/c no.:

* Bob 1000
* James 1001
* Bradley 1002
* Stephen 1003
* Ryan 1004

Program Features:

The program provides 2 types of users: a normal client and a manager.

Each class has different options. Both types of users have different unique login Ids - a username and password for the manager, and a pin number for the user. Text files are used to store data for clients and their account details.

Manager privileges:

A manager can register a new user, look up a client by entering their pin number and even get a list of all accounts stored in the program. Managers also carry administrative privileges having access to modify and delete customer information.

Customer privileges:

A customer can check the remaining balance, withdraw, and deposit money.

**Code:**

// main.cpp

// project

#include<iostream>

#include<cctype>

#include<iomanip>

#include <string>

#include <stdlib.h>

#include <fstream>

using namespace std;

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// CLASS USED IN PROJECT

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

class account

{

int acno;

char name[50];

int deposit;

char type;

public:

void create\_account(); //function to get data from user

void show\_account() const; //function to show data on screen

void modify(); //function to add new data

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() const; //function to show data in tabular format

int retacno() const; //function to return account number

int retdeposit() const; //function to return balance amount

char rettype() const; //function to return type of account

}; //class ends here

void account::create\_account()

{

cout<<"\nEnter the Account No. :";

cin>>acno;

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

cin.ignore();

cin.getline(name,50);

cout<<"\nEnter type of the account (C/S): ";

cin>>type;

type=toupper(type);

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

cin>>deposit;

cout<<"\nAccount Created.";

}

void account::show\_account() const

{

cout<<"\nAccount number: "<<acno;

cout<<"\nAccount holder Name: ";

cout<<name;

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

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

}

void account::modify()

{

cout<<"Account number: "<<acno;

cout<<"\nModify account holder name: ";

cin.ignore();

cin.getline(name,50);

cout<<"Modify type of account: ";

cin>>type;

type=toupper(type);

cout<<"Modify balance amount : ";

cin>>deposit;

}

void account::dep(int x)

{

deposit+=x;

}

void account::draw(int x)

{

deposit-=x;

}

void account::report() const

{

cout<<acno<<setw(10)<<" "<<name<<setw(10)<<" "<<type<<setw(6)<<deposit<<endl;

}

int account::retacno() const

{

return acno;

}

int account::retdeposit() const

{

return deposit;

}

char account::rettype() const

{

return type;

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function declaration

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

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\_withdraw(int, int); // function to desposit/withdraw amount for given account

void intro(); //introductory screen function

char uname [20];

char pass[5];

void newmanager();

void managerlogin();

void masterscreen();

void userscreen();

void managerscreen();

void secretscreen();

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// the MAIN FUNCTION OF PROGRAM

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// this is the first screen the user sees

int main()

{

int a ;

cout << "Login:";

cout <<"\n1 - Client Menu \n2- Manager Menu: \n3- Credits";

cout <<"\n\nSelect Your Option (1-3): ";

cin>>a;

if (a ==1)

{

userscreen();

}

if (a==2)

{

managerlogin();

}

if (a==3)

{

intro();

}

system("pause");

// unlisted - send user to a secret debugging menu,

// they need to sign in as a manager to view

if (a==4)

{

masterscreen();

}

return 0;

}

// Unlisted screen - this screen has all the options combined (used for debugging)

void masterscreen()

{

char ch;

int num;

do

{

system("cls");

cout<<"\nMAIN MENU";

cout<<"\n1. NEW ACCOUNT";

cout<<"\n2. DEPOSIT AMOUNT";

cout<<"\n3. WITHDRAW AMOUNT";

cout<<"\n4. BALANCE INQUIRY";

cout<<"\n5. ALL ACCOUNT HOLDER LIST";

cout<<"\n6. CLOSE AN ACCOUNT";

cout<<"\n7. MODIFY AN ACCOUNT";

cout<<"\n8. NEW MANAGER ACCOUNT";

cout<<"\n9. EXIT";

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

cin>>ch;

system("cls");

switch(ch)

{

case '1':

write\_account();

break;

case '2':

cout<<"\nEnter the account No. : "; cin>>num;

deposit\_withdraw(num, 1);

break;

case '3':

cout<<"\nEnter the account No. : "; cin>>num;

deposit\_withdraw(num, 2);

break;

case '4':

cout<<"\nEnter the account No. : "; cin>>num;

display\_sp(num);

break;

case '5':

display\_all();

break;

case '6':

cout<<"\nEnter the account No. : "; cin>>num;

delete\_account(num);

break;

case '7':

cout<<"\nEnter the account No. : "; cin>>num;

modify\_account(num);

break;

case '8':

newmanager();

break;

case '9':

cout<<"\nThank you for using the Bank Management System";

break;

default :cout<<"\a";

}

cin.ignore();

cin.get();

}

while(ch!='8');

}

//this creates a new manager user

void newmanager()

{

ofstream fi;

fi.open("managerusername.txt",ios::out|ios::app);

// check if the file exists

if( !fi.is\_open() )

{

cout << "Error opening file.";

exit(-1);

}

cout<<"\nEnter desired username: ";

cin.clear();

cin.ignore();

cin.getline(uname, 20);

cout<<"\nEnter desired password: ";

for(int i=0; i<5; i++)

{

pass[i]=getchar();

cout<<"\*";

}

cout << "\nThank you!";

cin.get();

fi << uname << "\n";

fi.write((char\*)&pass,sizeof(pass));

}

//this is the normal user screen

void userscreen()

{

char ch;

int num;

do

{

system("cls");

cout << "''''''''''''''''CLIENT MENU''''''''''''''''\n";

cout << "--------------------------------------------\n";

cout<<"1. Balance Enquiry";

cout<<"\n2. Withdraw Amount";

cout<<"\n3. Deposit Amount";

cout<<"\n4. EXIT";

cout<<"\nSelect Your Option (1-4): ";

cin>>ch;

system("cls");

switch(ch)

{

case '1':

cout<<"Enter the account No. : "; cin>>num;

display\_sp(num);

break;

case '2':

cout<<"Enter the account No. : "; cin>>num;

deposit\_withdraw(num, 2);

break;

case '3':

cout<<"Enter the account No. : "; cin>>num;

deposit\_withdraw(num, 1);

break;

case '4':

cout<<"\nThank you for using the Bank Management System!";

break;

default :cout<<"\a";

}

cin.ignore();

cin.get();

}

while(ch!='4');

}

//this is the manager's screen

void managerscreen()

{

char ch;

int num;

do

{

system("cls");

cout << "\n''''''''''''''''MANAGER MENU''''''''''''''''\n";

cout << "--------------------------------------------\n";

cout << "Welcome back Manager!\n";

cout << "Please Choose the appropriate number from \n";

cout << "following options to continue\n\n";

cout << "1. Register New User\n";

cout << "2. Get Customer List\n";

cout << "3. Modify Customer Details\n";

cout << "4. Delete Account\n\n";

cout << "Please enter your choice number (1 - 4):\n";

cin >> ch;

switch (ch) {

case '1':

write\_account();

break;

case '2':

display\_all();

break;

case '3':

cout<<"\nEnter the account No. : "; cin>>num;

modify\_account(num);

break;

case '4':

cout<<"\nEnter the account No. : "; cin>>num;

delete\_account(num);

break;

case '5':

cout<<"\nThank you for using the Bank Management System!";

break;

default:cout<<"\a";

}

cin.ignore();

cin.get();

}

while(ch!='4');

}

//the login page for the manager user

void managerlogin()

{

char name2[20], pass2[6];

int found = 0;

cout << "Enter username: ";

cin.clear();

cin.ignore();

cin.getline(name2,20);

cout << "Enter password: ";

for(int j =0; j<5; j++)

{

pass2[j]=getchar();

cout<<"\*";

}

pass2[5] = 0;

ifstream fin;

fin.open("", ios::in);

fin.seekg(0);

fin.getline(uname,20);

if(strcmp(name2,uname)!=0)

{

cout <<"\nNo such user\n";

return;

}

while(!fin.eof())

{

fin.read((char\*)&pass, sizeof(pass));

if(strcmp(pass2,pass)==0)

found = 1;

}

if (found ==1)

managerscreen();

else

cout<<"\nSorry!\n";

fin.close();

}

//unlisted screen - manager needs to sign in to view the master screen (used for debugging)

void secretscreen()

{

char name2[20], pass2[6];

int found = 0;

cout << "Enter username: ";

cin.clear();

cin.ignore();

cin.getline(name2,20);

cout << "Enter password: ";

for(int j =0; j<5; j++)

{

pass2[j]=getchar();

cout<<"\*";

}

pass2[5] = 0;

ifstream fin;

fin.open("managerusername.txt", ios::in);

fin.seekg(0);

fin.getline(uname,20);

if(strcmp(name2,uname)!=0)

{

cout <<"\nNo such user\n";

return;

}

while(!fin.eof())

{

fin.read((char\*)&pass, sizeof(pass));

if(strcmp(pass2,pass)==0)

found = 1;

}

if (found ==1)

masterscreen();

else

cout<<"\nSorry!\n";

fin.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to write in file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void write\_account()

{

account ac;

ofstream outFile;

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

ac.create\_account();

outFile.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

outFile.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to read specific record from file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void display\_sp(int n)

{

account ac;

bool flag=false;

ifstream inFile;

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

if(!inFile)

{

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

return;

}

cout<<"\nBalance details:\n";

while(inFile.read(reinterpret\_cast<char \*> (&ac), sizeof(account)))

{

if(ac.retacno()==n)

{

ac.show\_account();

flag=true;

}

}

inFile.close();

if(flag==false)

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

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to modify record of file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void modify\_account(int n)

{

bool found=false;

account ac;

fstream File;

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

if(!File)

{

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

return;

}

while(!File.eof() && found==false)

{

File.read(reinterpret\_cast<char \*> (&ac), sizeof(account));

if(ac.retacno()==n)

{

ac.show\_account();

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

ac.modify();

int pos=(-1)\*static\_cast<int>(sizeof(account));

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

File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

cout<<"\nRecord Updated.";

found=true;

}

}

File.close();

if(found==false)

cout<<"\nRecord Not Found.";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to delete record of file

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void delete\_account(int n)

{

account ac;

ifstream inFile;

ofstream outFile;

inFile.open("account.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(reinterpret\_cast<char \*> (&ac), sizeof(account)))

{

if(ac.retacno()!=n)

{

outFile.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

}

}

inFile.close();

outFile.close();

remove("account.dat");

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

cout<<"\nRecord Deleted.";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to display all accounts deposit list

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void display\_all()

{

account ac;

ifstream inFile;

inFile.open("account.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(reinterpret\_cast<char \*> (&ac), sizeof(account)))

{

ac.report();

}

inFile.close();

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// function to deposit and withdraw amounts

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void deposit\_withdraw(int n, int option)

{

int amt;

bool found=false;

account ac;

fstream File;

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

if(!File)

{

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

return;

}

while(!File.eof() && found==false)

{

File.read(reinterpret\_cast<char \*> (&ac), sizeof(account));

if(ac.retacno()==n)

{

ac.show\_account();

if(option==1)

{

cout<<"\n\nTo deposit amount:";

cout<<"\nEnter the amount to be deposited: ";

cin>>amt;

ac.dep(amt);

}

if(option==2)

{

cout<<"\n\nTo withdraw amount:";

cout<<"\nEnter the amount to be withdrawn: ";

cin>>amt;

int bal=ac.retdeposit()-amt;

if((bal<25 && ac.rettype()=='S') || (bal<25 && ac.rettype()=='C'))

cout<<"\nInsufficient balance!";

else

ac.draw(amt);

}

int pos=(-1)\*static\_cast<int>(sizeof(ac));

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

File.write(reinterpret\_cast<char \*> (&ac), sizeof(account));

cout<<"\nRecord Updated.";

found=true;

}

}

File.close();

if(found==false)

cout<<"\nRecord Not Found. ";

}

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

// INTRODUCTION FUNCTION

//\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

void intro()

{

cout<<"\nBANK MANAGEMENT SYSTEM";

cout<<"\nMade by:Bradley Turner";

cout<<"\nSchool: Stony Brook University\n\n";

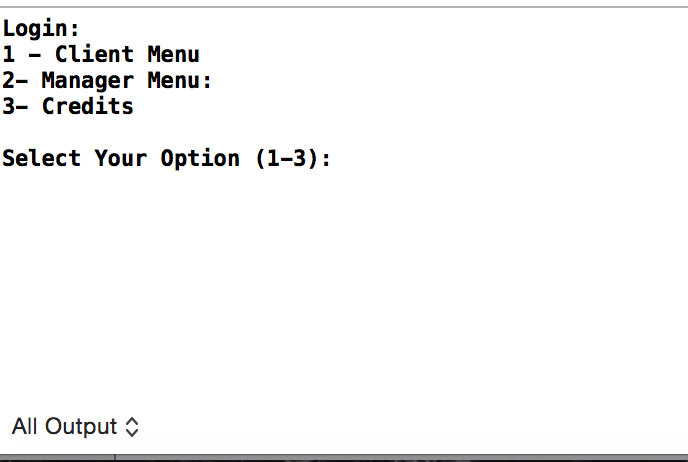
cin.get();

}

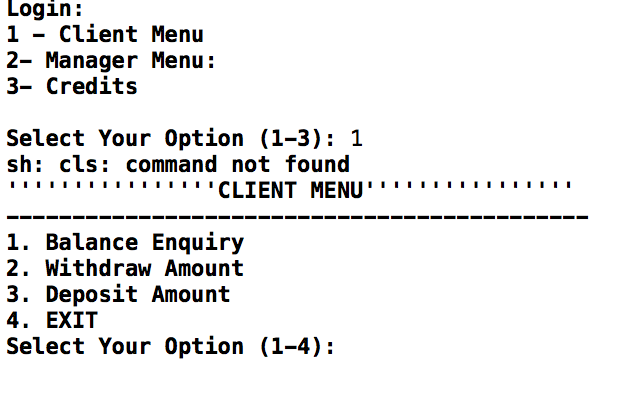
**Output Windows Demonstrating the Functions of the Program:**

\*Please note that the client accounts used in these screenshots are not the same as the ones submitted within the exe file

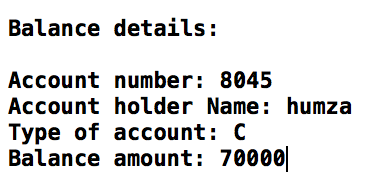
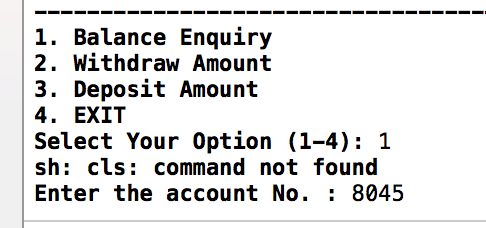
Home page:



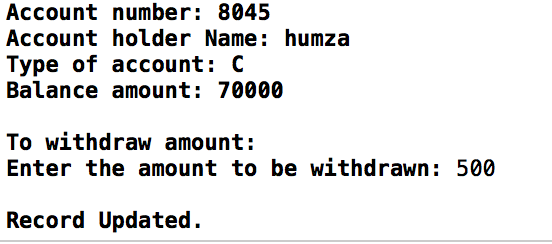
**Client menu:**



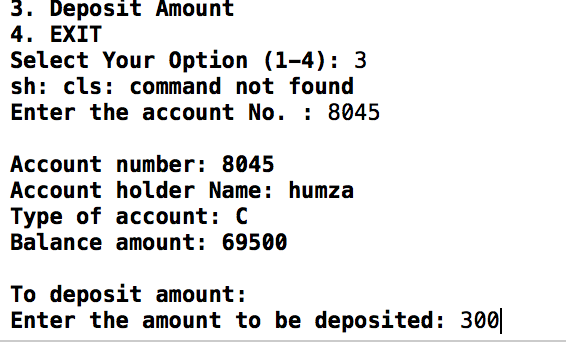
**Balance enquiry:**



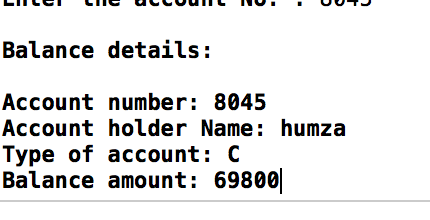
**Withdraw:**

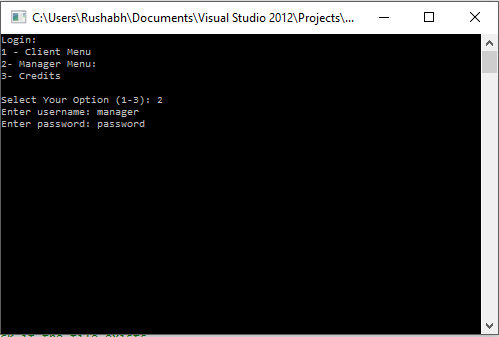


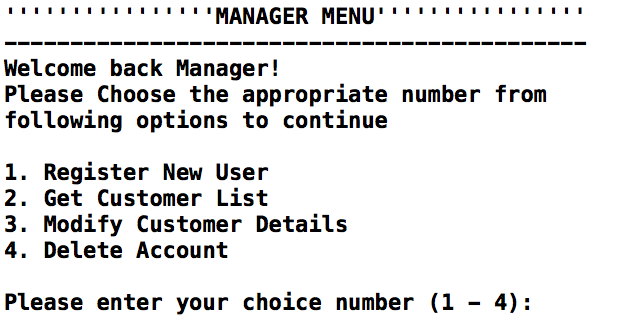
**Deposit:**



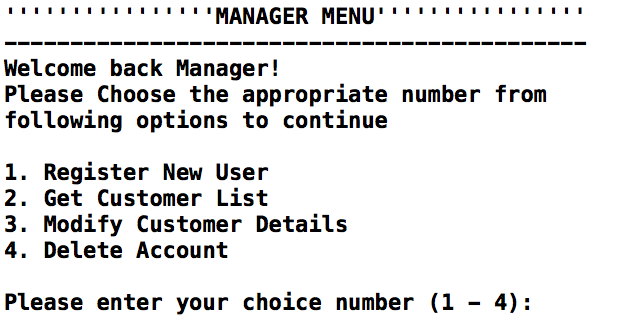
**New Balance:**

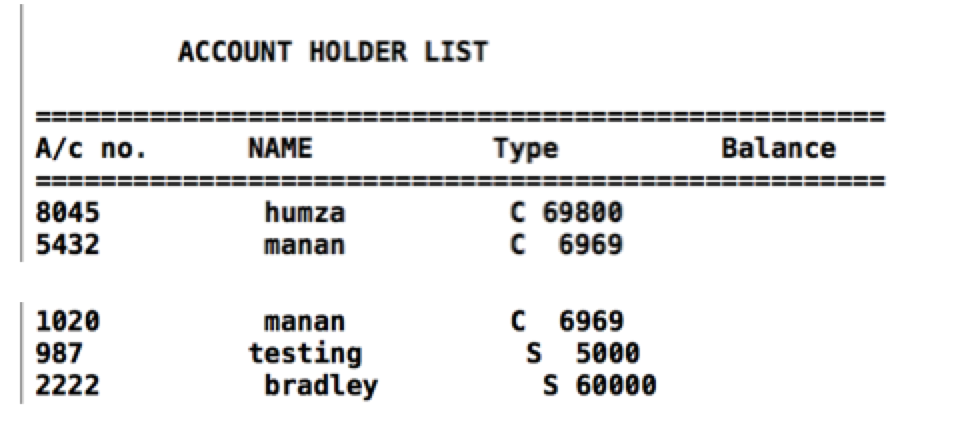


**Manager Options:**

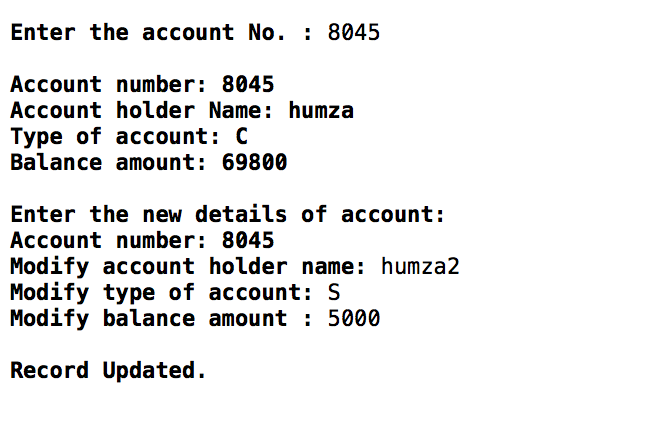


**Register new user:**

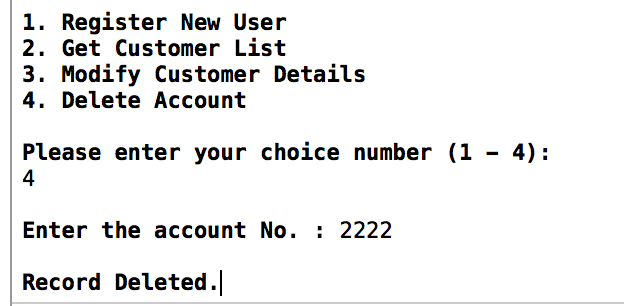


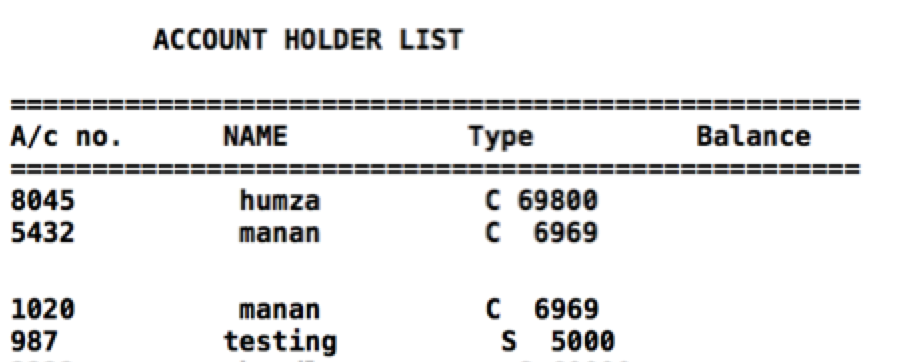
**Get customer list:** 

**Modify customer details:**



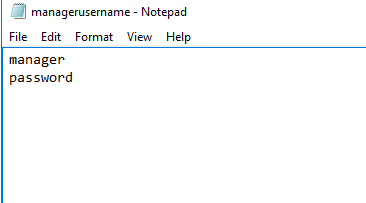
**Deleting a record:**



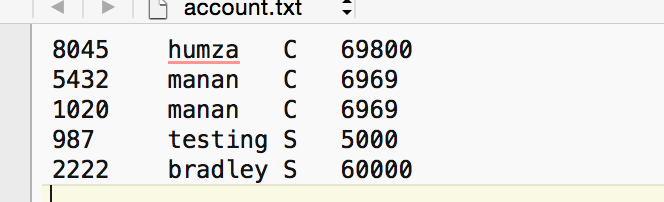
**Record list check:**

**Text files:**

**Manager credentials:**



**Client credentials:**



**In All:**

For this project, we used concepts of C++ programming we learned throughout the year in ESE 224. In order for this project to come together we had to pitch in our own ideas and share the workload evenly so the project would be completed smoothly. From doing this project, we not only further familiarized ourselves with many concepts in the C++ programming language, as well as how to efficiently cooperate as a team and delegate tasks.