International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)
Impact Factor: 5.22 (SJIF-2017), e-ISSN: 2455-2585
Volume 5, Issue 01, January-2019

MODERN CLIENT-SERVER BANK MANAGEMENT SYSTEM

Dr. Darshankumar C. Dalwadi[1], Ronit Rout[2]

[1]*EC Department, BVM Engineering College.*
[2]*EC Department, BVM Engineering College.*

Abstract— The Bank Account Management System is an application for maintaining a person's account in a bank. In this paper we have present the working of a banking account system and cover the basic functionality of a Bank Account Management System. To develop a project for solving financial applications of a customer in banking environment in order to nurture the needs of an end banking user by providing various ways to perform banking tasks. Also to enable the user's work space to have additional functionalities which are not provided under a conventional banking project. The proposed bank Account Management System is based on relevant technologies. It also contains advanced C++ graphics to make it more appealing. We will be providing password encryption on the server side as well for enhanced security to avoid data theft.

Keywords— Banking, Encryption, Client- Server, C++, Socket Programming, DEV C++ IDE.

I. INTRODUCTION

In this paper, we have present the client-server based banking management system which is designed in DEV IDE and programming in C++ along with the basics of socket programming for connectivity between the servers and various clients. The server and client can be connected through IP addresses. We have install the server and client on the same computer and connect them using Local Host Loopback IP address. The server can also be loaded into one system and the clients into multiple systems and connected together using Local Area Network (LAN).The server can then be put into "passive open" mode by entering 0.0.0.0 as the IP address. The server then can listen to any client, which is in "active open" mode connected in the LAN by entering the respective IP address.

The banking system is to provide basic functionalities like opening account, deleting account, debit, credit and third party transfers etc. along with password encryption on server side for enhanced security. It contains advanced C++ graphics to make it more user friendly and convenient.

The benefit of client-server system is that even if we completely close the program and then run the program again and login into our account our account will still be present and we can do the transactions. So it can be used multiple number of times.

II. SYSTEM MODEL

In this section we have discussed the system model of client-server application with database. We have also discussed the flowchart of the client-server application involving sockets.

A. *Client-Server Application System Model*

Figure 1 shows the basic client server application with database model [1,2,3]. The client-server management system is widely used in today's world. In this system, firstly the server is configured in a "passive open" mode. It means that the server is ready to connect to any client in any network. The client which is in "active open" mode, now gets connected to the server by entering the IP address[4,5,6]. After the request is sent to the server, it sends this data to the database to confirm the essential credentials. After confirming, the database server sends the result to the server. The server then searches for the information which the client is seeking for and then sends the data to the client.

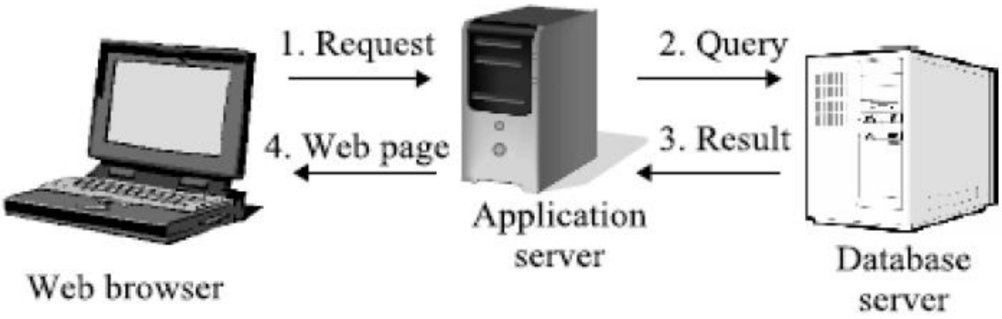


Fig. 1 System Model of a Client Server Application with Database

B. *Client-Server Application Flowchart*

Figure 2 shows the flowchart of client-server application with socket involvement [7,8]. The client first creates socket and then binds sockets to port. Then it sends the connect request to server and waiting server to respond. This is called "active open" mode. At the other end, the server creates a socket and binds socket to port. It continuously listens to the clients. This is called "passive open" mode [9,10]. The client traffic slowly lines up in the buffer to be accepted by the server. Finally, a client is able to connect to the server. Once the connection is established, they can simultaneously transmit and receive information. After this is completed, the connection is terminated. Then the request from other client is accepted and it is served.

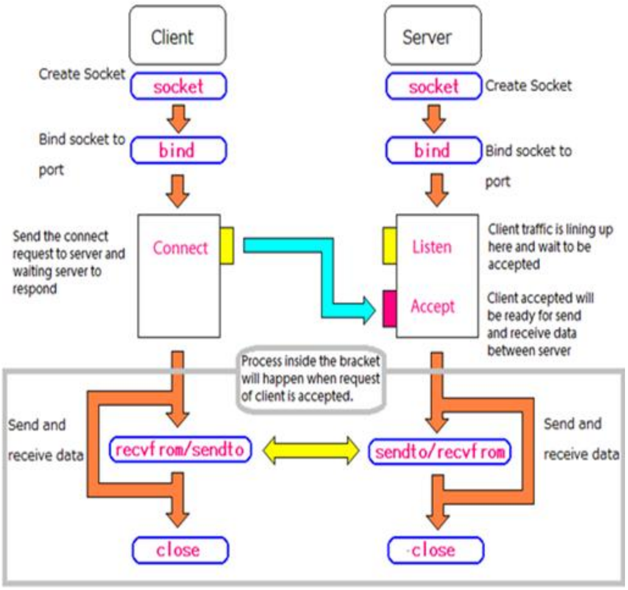

Client
Server
Create Socket
socket
socket
Create Socket
Bind socket to port
bind
bind
Bind socket to port
Send the connect request to server and waiting server to respond
Connect
Listen
Client traffic is lining up here and wait to be accepted
Accept
Client accepted will be ready for send and receive data between server
Process inside the bracket will happen when request of client is accepted.
Send and receive data
recvfrom/sendto
sendto/recvfrom
Send and receive data
close
close

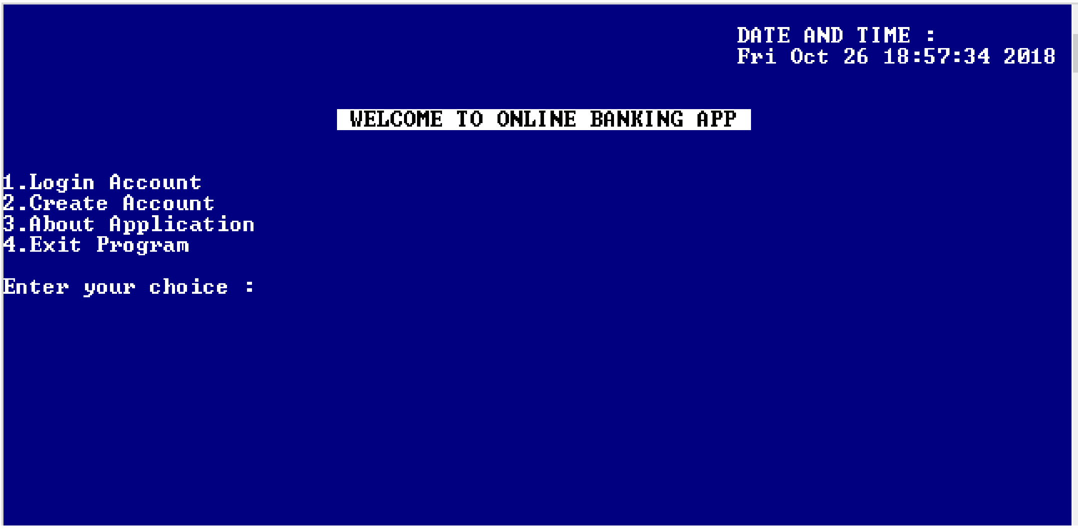
Fig. 2 Flowchart of client server application involving sockets

III. SIMULATION RESULTS

In this section, we have discussed the various simulation results at client as well as server side. We have shown the demonstration of the software at client and server sides.

A. *Simulation output at Client side*

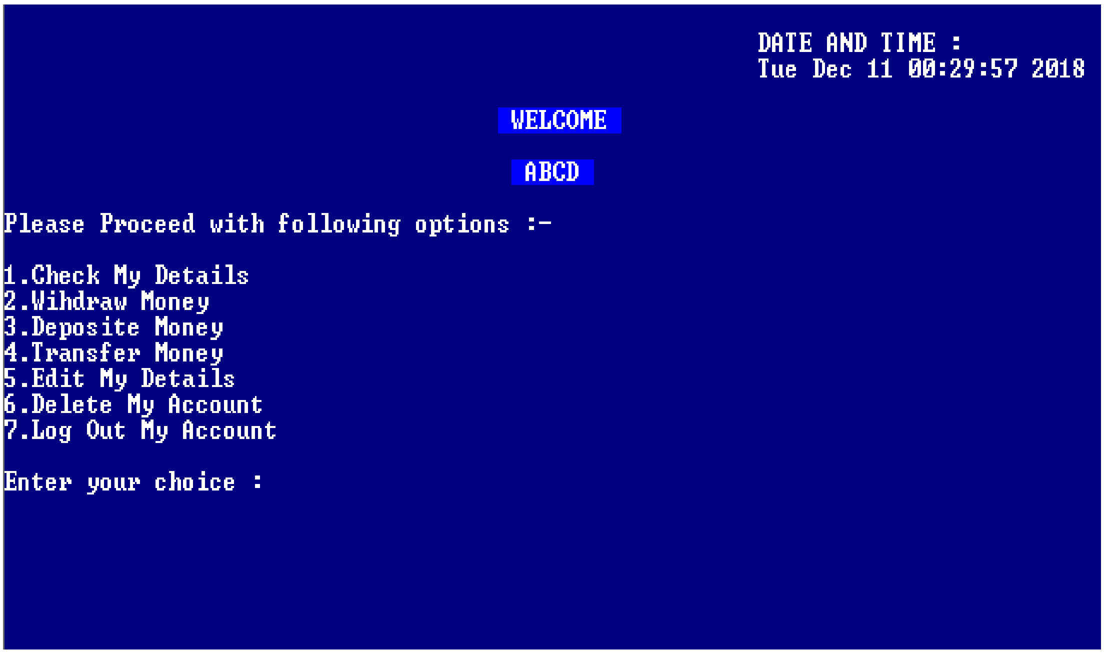
Figure 3 shows the client window. It contains a welcome message to the app for the users. It contains date and time at the upper right corner. It contains basic operations like creating new account, logging into account, information about application and one option to exit the program and then it asks the user to enter their choice.



```
                                                        DATE AND TIME :
                                                        Fri Oct 26 18:57:34 2018

                         WELCOME TO ONLINE BANKING APP


1.Login Account
2.Create Account
3.About Application
4.Exit Program

Enter your choice :
```

Fig. 3 Opening Window of Application (Client Side)

Figure 4 shows the window comes after entering the correct credentials and getting logged in successfully at the client side. It consists of date and time at upper right corner along with the name of account holder at center. It consists of basic banking functionalities like checking the details; withdraw money, deposit money, transfer money, editing details, deleting account permanently and logging out option. Then it asks for the user choice.

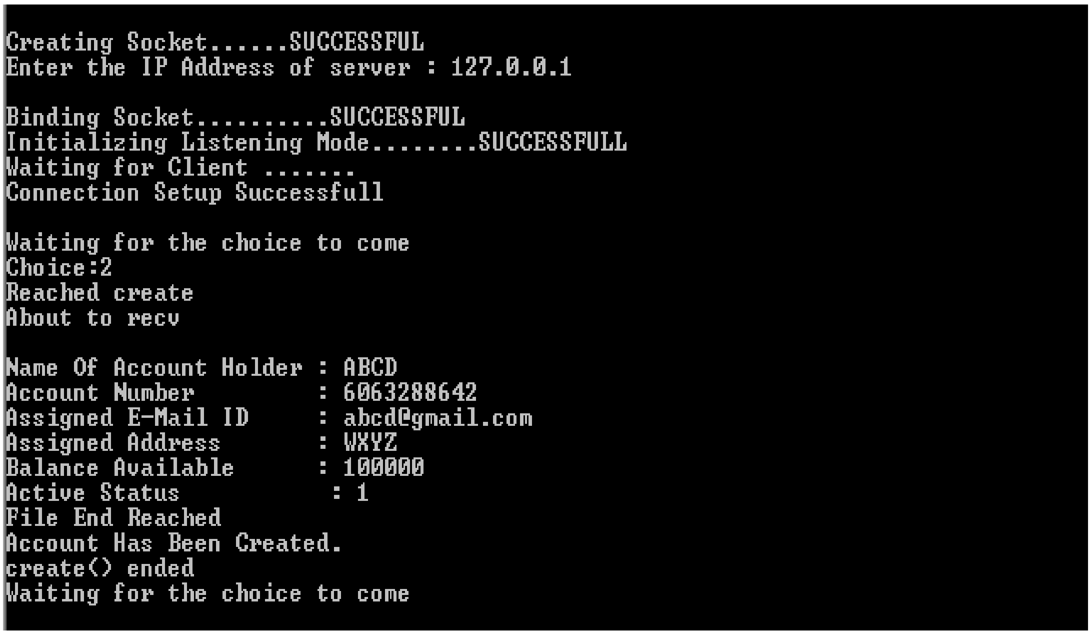


```
                                                       DATE AND TIME :
                                                       Tue Dec 11 00:29:57 2018

                                    WELCOME

                                     ABCD

Please Proceed with following options :-

1.Check My Details
2.Wihdraw Money
3.Deposite Money
4.Transfer Money
5.Edit My Details
6.Delete My Account
7.Log Out My Account

Enter your choice :
```

Fig. 4 Window after logging in successfully

B. *Simulation output at Server Side*

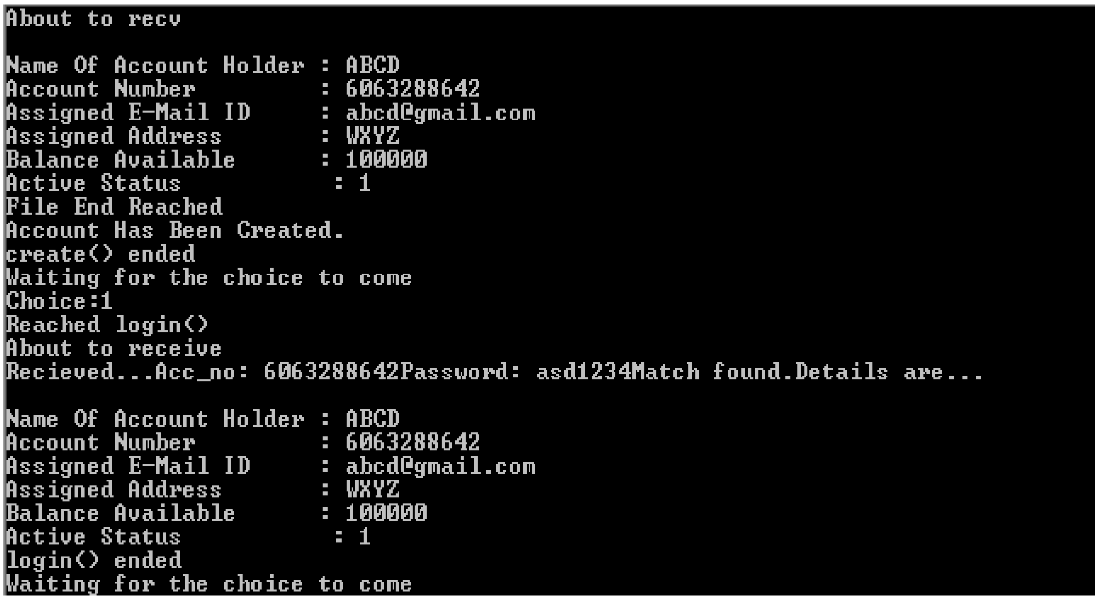
Figure 5 shows the simulation output at server side. First the socket creation is successful. In this the client and server are in the same system and that is the reason we used the loopback IP address (127.0.0.1). The socket is binded and then it waits for the client to get connected to it. Once the connection is successful, it waits for the choice from the user. In this case, the choice entered is 2(Create Account). The information is entered at the client side. A random 10 digit account number is allocated to the client. The data entered by the client, except password, is reflected at the server end and is stored in a file "BANK".



```
Creating Socket......SUCCESSFUL
Enter the IP Address of server : 127.0.0.1

Binding Socket..........SUCCESSFUL
Initializing Listening Mode........SUCCESSFULL
Waiting for Client .......
Connection Setup Successfull

Waiting for the choice to come
Choice:2
Reached create
About to recv

Name Of Account Holder : ABCD
Account Number         : 6063288642
Assigned E-Mail ID     : abcd@gmail.com
Assigned Address       : WXYZ
Balance Available      : 100000
Active Status           : 1
File End Reached
Account Has Been Created.
create() ended
Waiting for the choice to come
```

Fig. 5 Server side window after successful creation of account and account number generation

Figure 6 also shows the window at the server end. In this case the user has entered option 1 (Login Account) at the client side. And then the user enters the account number and password. This account number and password is checked in the database and if they match and are valid, the server displays "Match Found" and then displays the information and then we are able to do all functionalities.



```
About to recv

Name Of Account Holder : ABCD
Account Number         : 6063288642
Assigned E-Mail ID     : abcd@gmail.com
Assigned Address       : WXYZ
Balance Available      : 100000
Active Status           : 1
File End Reached
Account Has Been Created.
create() ended
Waiting for the choice to come
Choice:1
Reached login()
About to receive
Recieved...Acc_no: 6063288642Password: asd1234Match found.Details are...

Name Of Account Holder : ABCD
Account Number         : 6063288642
Assigned E-Mail ID     : abcd@gmail.com
Assigned Address       : WXYZ
Balance Available      : 100000
Active Status           : 1
login() ended
Waiting for the choice to come
```

Fig. 6 Window after logging into account (the details are checked in database) and displayed at server end

IV. CONCLUSIONS

We have a dynamic looking and an interactive Banking Management System. We have implemented encryption of password at the server end to provide more security. We have further explored the likes of advanced C++ and Windows Socket Programming and have good command over them at the end of the project. The GUI can further be enhanced by using JAVA or PYTHON. It can be used in making payments if security is enhanced and linked to bank or for digital currencies. It can further be extended to system like Library management systems, Canteen management systems and Attendance systems. For more security, encryption algorithms can be used. The databases like mySQL or MongoDB or Oracle can also be used to store the data in an organized manner.

REFERENCES

[1] S. M. Metev and V. P. Veiko, Laser Assisted Microtechnology, 2nd ed., R. M. Osgood, Jr., Ed. Berlin, Germany: Springer-Verlag, 1998.

[2] J. Breckling, Ed., The Analysis of Directional Time Series: Applications to Wind Speed and Direction, ser. Lecture Notes in Statistics. Berlin, Germany: Springer, 1989, vol. 61.

[3] S. Zhang, C. Zhu, J. K. O. Sin, and P. K. T. Mok, “A novel ultrathin elevated channel low-temperature poly-Si TFT,” IEEE Electron Device Lett., vol. 20, pp. 569–571, Nov. 1999.

[4] M. Wegmuller, J. P. von der Weid, P. Oberson, and N. Gisin, “High resolution fiber distributed measurements with coherent OFDR,” in Proc. ECOC’00, 2000, paper 11.3.4, p. 109.

[5] R. E. Sorace, V. S. Reinhardt, and S. A. Vaughn, “High-speed digital-to-RF converter,” U.S. Patent 5 668 842, Sept. 16, 1997.

[6] FLEXChip Signal Processor (MC68175/D), Motorola, 1996.

[7] “PDCA12-70 data sheet,” Opto Speed SA, Mezzovico, Switzerland.

[8] A. Karnik, “Performance of TCP congestion control with rate feedback: TCP/ABR and rate adaptive TCP/IP,” M. Eng. thesis, Indian Institute of Science, Bangalore, India, Jan. 1999.

[9] J. Padhye, V. Firoiu, and D. Towsley, “A stochastic model of TCP Reno congestion avoidance and control,” Univ. of Massachusetts, Amherst, MA, CMPSCI Tech. Rep. 99-02, 1999.

[10] Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specification, IEEE Std. 802.11, 1997.