Department of Mathematics and Humanities

CS303 - Computer Networks

Client-Server Application-Automated Teller Machine (ATM)

Author:
Tarang Pansuriya
Kunjera Chetan
Dharmik Patel
Jatin Dhanani

Table of Contents

1 Introduction of Group members

Pansuriya Tarang Bharatbhai	**I20MA005**
Kunjera Chetan Dayabhai	**I20MA018**
Dharmik Patel	**I20MA020**
Dhanani Jatin Chimanbhai	**I20MA022**

2 Application details

An automated teller machine (ATM) is a computerized telecommunications device that provides clients with access to financial transactions in a public space without the need for a human clerk or bank teller.This application allows users to connect to a server in order to access their bank account information and allow user to perform some operations . This information can include account balances and operations like changing pin and debit money.

Client to Server
A client-server application is a program that runs on a client computer and accesses a server. The client-server model is a distributed application structure that partitions tasks or workloads between the providers of a resource or service, called servers, and service requester, called clients.

Codes-Algorithms-Flowcharts
We use python and leverage the help of sockets to create a client to server network. The server program contains all the details of particular client and there is a dedicated files which keep tracks of account balance and pin of the particular user. The server side of the program and the client part of the program are in different files but part of the same program. 'client.py' file contains the code for the client side of the application. It handles the communication with the server and the display of information to the user. We would first go through our server side of the program, then the client side of the program.

Server.py

- This is a python script which enables server to access information of client, it sets up a TCP/IP connection between the two machines and allows them to exchange information in real time.
- We first created a socket and made connection between the client and server via passing the same port number.

```
from socket import *
s=socket(AF_INET,SOCK_STREAM) # making socket with TCP connection
s.bind(('',11846))  # connecting client and server via particular port
s.listen(1)                   # allowed number of client
```

- After making successful connection we accessed pin(pin:1234) and account balance of the current user from external file.
- Now, server is ready with all the required information about a client. So, it will wait till client approaches and tried to make connection.

```
while True:
conne,addr=s.accept()
conne.send('hey client! Please enter your Pin!'.encode())
```

- As soon as client make connection with server, the client will be asked to enter pin number to access account. In case of wrong pin, client will be asked to retype the pin.

```
while int(p)!=pin:
    conne.send('False'.encode())
    p=conne.recv(1024).decode()
if int(p)==pin:
    conne.send('True'.encode())
```

- After verifying the pin, the server will be given request to perform certain operation. This request will be sent by client by sending particular string.
- There are 4 possible operation which server can perform here.
  1. Change Pin
  2. Check Balance
  3. Debit Money
  4. Credit Money

Client.py

- In client.py, we have created socket object and assigned it to variable cli. Then we have connected to the server using cli.connect((HOST, PORT)).
- Client side if the application contains all the codes related to what a user interface will look like.
- Client side has 'while loop' which will keep running till we input correct pin.
- Client side will interact with user and carry forward the operation request to server. The server in return perform that operation and send the response to client side again as a conformation.

- When user will choose the 'Exit' operation, only client side program will terminate and server will stay on. So, that client can make connection again if user requires.

How to access application

1. Use this Link to access all the files related to code.
2. Since this is TCP based application, make sure to run 'server.py' first and then 'client.py'.

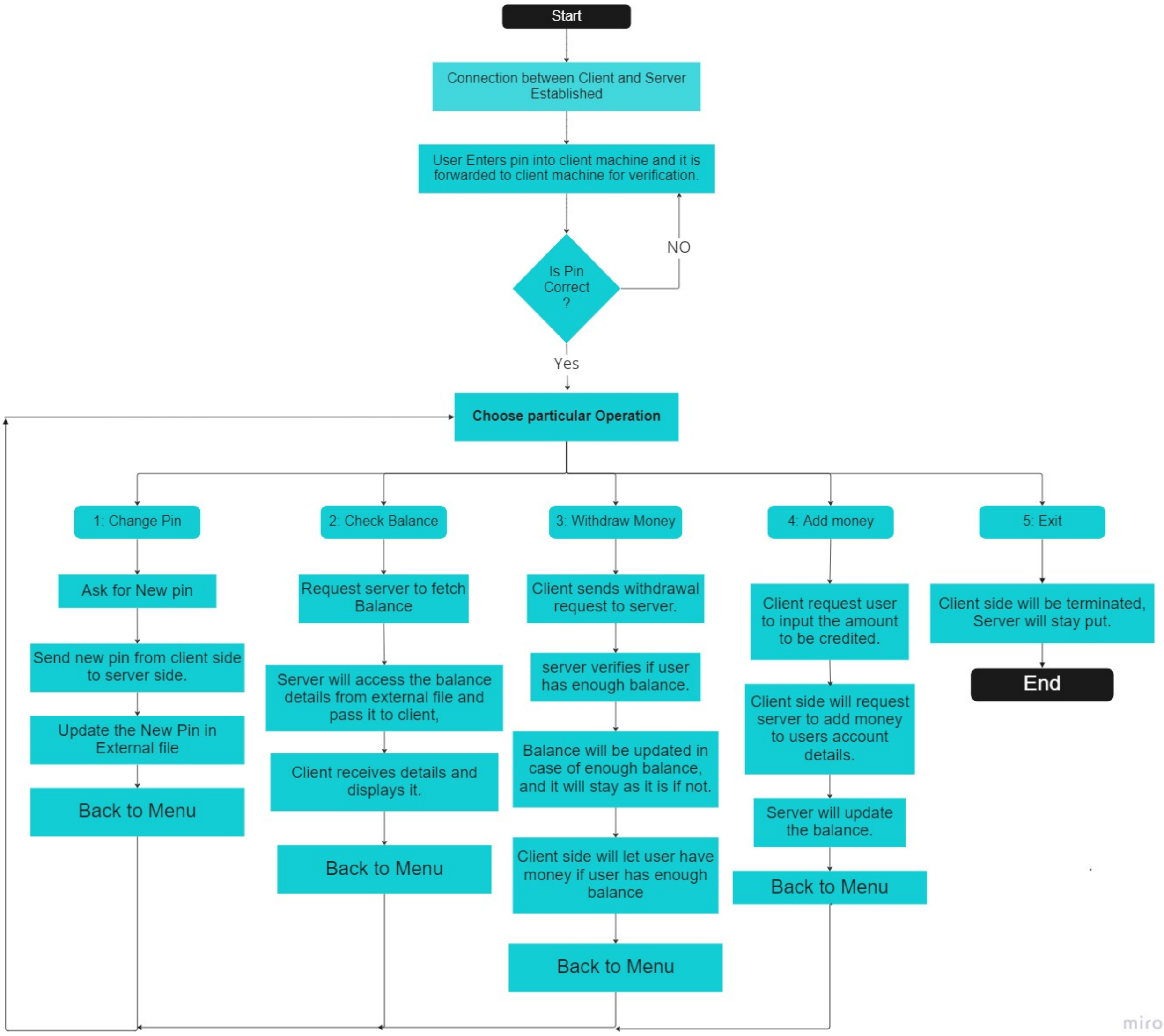
Start
Connection between Client and Server Established
User Enters pin into client machine and it is forwarded to client machine for verification.
Is Pin Correct ?
NO
Yes
Choose particular Operation
1: Change Pin
2: Check Balance
3: Withdraw Money
4: Add money
5: Exit
Ask for New pin
Send new pin from client side to server side.
Update the New Pin in External file
Back to Menu
Request server to fetch Balance
Server will access the balance details from external file and pass it to client,
Client receives details and displays it.
Back to Menu
Client sends withdrawal request to server.
server verifies if user has enough balance.
Balance will be updated in case of enough balance, and it will stay as it is if not.
Client side will let user have money if user has enough balance
Back to Menu
Client request user to input the amount to be credited.
Client side will request server to add money to users account details.
Server will update the balance.
Back to Menu
Client side will be terminated, Server will stay put.
End
miro

Figure 1: This is a Flowchart of our application

Important Note
In our server side of the code we are accessing external file to keep track of the code.

```
with open(r'C:\Users\Dharmik\Desktop\Python Projects\client server\pin.txt', 'r') as f:
#be sure to change this location of the file.
    lines = f.read().splitlines()
    last_line = lines[-1]
    pin=int(last_line)
```

In the above code we are using file handling to access the 'txt' file from our local directory. So, In order to run the code without any error,***it is important to change the location of the file according to the location of 'txt' in their system.***

3 Conclusion

This client-server program of Automated Teller Machine is working in smooth manner without any error.After ending every session we can observe the changes in detail in external file which has been made by the user. This program can still excel by making it scalable, i.e. we can provide service to more users.