**PROJECT PROBLEM:**

Implement software for the next generation of bank ATM machines using the protocol that was chosen in class. The functionality include creating account, deposit, withdraw, verification of action, balance, show n transactions, buy stamps and terminate transaction.

**APPROACH:**

The protocol defines how the data is send from the client to the server. We will have to create a buffer that is big enough to hold the data. The first number is the transaction number followed by a single space and rest of the data separated by a space.

**SOLUTION:**

We established a connection between client and the server. We defined a port number where the client and the server can read and write. We get the server up and running and it’s listening to the port we defined before. The server side loads the data from the files and stores the user data in a doubly linked list. The protocol defines what command is send to the server and back to the client. We defined a string buffer that is big enough to hold all data. The data is formatted where the first number identifies the transaction. We take the input from the user and store the data the in the string where each data separated by a space. The client initiates the request to the server. The server reads the buffer and converts the first value to an integer by using function call ATOI that converts a string to an integer. Then it checks to see if the remaining data is present. If any data is missing then an error code will be send to the client. If the all the data is present the server side will call the appropriate function using the protocol number.

We setup SSN as the unique since there can be only one user with same SSN. We also check for the unique combination of first name and PIN to see there are no duplicate combinations when creating a new user. Whenever the user is withdrawing the amount is decreased from the ATM, which has initial of $50000.00 to start with. The ATM can hold up to 1000 stamps costing $1 per stamp. Whenever the user deposits the amount it’s added to the deposit amount of the ATM, which is $0.00 to start off with. The ATM can’t make any more deposits that exceed the deposit amount over $50000.00. That means the maximum ATM can hold at one time is $100000.00. Whenever the user purchase stamps the amount is withdrawn from the users account and the number of stamps is decreased from the ATM.

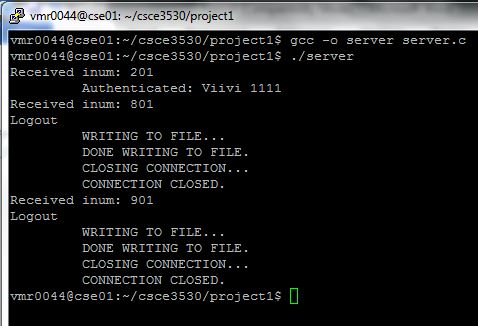
Transactions for each user are kept in array of 5. We keep track of which index we are currently on and work backwards to get the last 5 transactions. The starting balance of each user is $0.00. This is increased as the user deposits money into the account. After the user successfully login the SSN is returned and stored on the server side until the user logs out. This SSN will be used for any future transaction such as finding the balance. When the user logs out the SSN will be assigned the default value of a negative integer.

If the failed attempts exceed 10 for a user then it locks the user from logging in. The failed attempts are stored for the user. User can’t unlock it unless an administrator resets the failed account for the user. We added a command to shut down the program on the client side. This is normally not present for an ATM to bank transaction but for our purposes we want to make sure client and the server side shuts down properly. The server side stops listing and open the port back up for other users and shuts down the program.

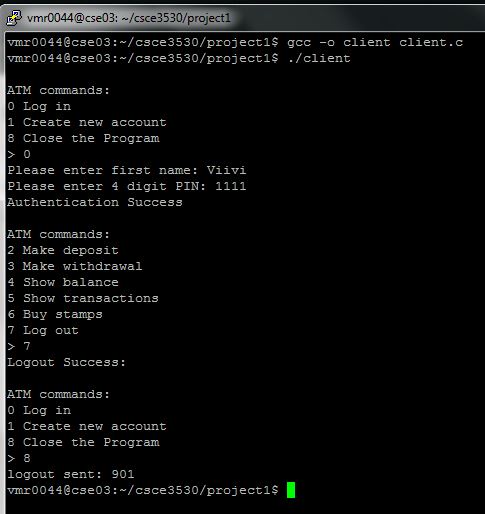
**SCREENSHOTS:**

The screenshots were obtained by testing the client and server program on various CSE machines.

Compiling and running the server (server.c) on the bank’s computer (cse01). Feedback for user authentication, log out, and close system.



Compiling and running the client program (client.c) on the ATM machine (cse03). User (fname: Viivi, pin: 1111) logs in, logs out, and the then the system is closed.



Further interaction between user (fname: fname, pin: 1235). User displays balance, makes a deposit of $10, then purchases 3 stamps, shows the last 2 transactions, then logs out.

Client:

|  |  |
| --- | --- |
| C:\Users\Viivi Rai\Documents\Viivi's Documents\UNT\2015_Spring_CSCE3530\Project1\final\screenshots\fname_client_01.JPG | C:\Users\Viivi Rai\Documents\Viivi's Documents\UNT\2015_Spring_CSCE3530\Project1\final\screenshots\fname_client_02.JPG |

Server:

