

CS341 / Assignment 4

Due Date: Tuesday 10/24/2017 @ 11:59 PM PST

/ Instructions

Please compress your code and README.txt file into a zip file or tar.gz.

/ Programming Assignment

Download the template Java program in Canvas: Files/ATMServer.zip

For this assignment you will create your own application layer protocol for a server that acts as a command and control server for an ATM network.

The Server (50% of your grade)

This server will be what ATMs communicate with. As such, you need to implement the following functionality in your protocol:

- Login – the client will pass you a user name and PIN, and your server will generate a token and pass it back to them if successful
- Logout – the client will pass you a token, and you will log them out
- Balance – The client will pass you a token and you will pass the balance of the account
- Withdraw – The client will pass you a token and an amount to withdraw, if the amount is less than the balance, you will pass back success
- Deposit – The client will pass you a token and an amount of money to deposit into the user's account

Additionally, you need to think of and handle all possible error situations (such as: bad username-PIN combinations and trying to withdraw more money than what you have).

DON'T PANIC!

I already implemented the database for you (in AccountsDB.java) which does not need to be modified. However, you do need to read the code, and familiarize yourself with how it works (a good place to start is AccountDB::main – it demonstrates how the basic database operations work... you can also compile the java file and run “java AccountDB” to see the output).

Out of the box, you can compile ATMServer.java by using “javac ATMServer” (or “make” if you use the Makefile), then run it by typing “java ATMServer”. In another window you can run “telnet 127.0.0.1 8080” and connect to your server. Type a line of text, and press enter. You see that the ATMServer printed your message to the screen. Additionally, the server also sends the text back to the telnet client and then closes the connection.

Another decision you must make is whether to keep the connection open between calls, or close the socket connection after each command. This is entirely up to you.

In addition to implementing your protocol in the ATMServer class (again, please look for the TODO comments), you MUST create your own custom client to connect to the server from scratch.

The Client (50% of your grade)

The client you will create (please name the class ATMClient) will use your new application layer protocol to communicate with your ATMServer.

This client should be a text application that functions similarly to a conventional ATM. It should ask the user for a username and PIN. Then, it should display a menu and allow the user to select commands. The client should also print user friendly messages.

/ Tips

- Start early!
- Ask questions... ask them in the discussion on Canvas first before asking me.
- Help other students! Don't give them your protocol, but feel free to collaborate via Canvas – I will look at the discussions at the end of the semester and students who help other students may get the benefit of the doubt if your grade is teetering on being a better grade
- The IP address of your server is 127.0.0.1
- The port of your server is 8080
- Make sure your server is running before you try to connect to it
- Install / use telnet to test out your server
- Only after you have your server working properly work on building the client
- There is a lot of code in your HTTP client from Assignment 1 that can be reused for building the ATM client
- I have provided you a basic Makefile... this makes things simpler when compiling your program

/ Extra credit

- Alter the Makefile to build your client as well as the server
- Make a graphical client
- Use SSL
- In general, build something awesome that goes above and beyond and I'll give you extra credit