COMP3331 Report for Assignment

Yifan WANG z5175023

Assignment Requirement Overview

This assignment requires to write a simple contact tracing app with a server and multiple client. The client will do the user authentication, tempID generation and contact log checking, the client will provide the user interface prompt and allow user to beacon message to other users (through another client). The server should block user with three unsuccessful attempts (with username and password case-sensitive) for 60s and client will not able to login during this time even on another IP address.

Student Production Details

This assignment is implemented by Java language (with Java 11) on MacOS, it was also been tested on CSE too (with VLab). This production implemented features include register, login, authentication, block, tempID generation, download tempID, upload contact logs and contact log checking. Please note that it did not implement peer to peer communication protocol (beaconing), so please test upload contact logs and contact log checking with a static contact log file (file name as "<z5175023>_contactlog.txt"), or manually create a log file by random selecting some tempIDs from tempIDs' file.

The server and clients are communicating base on the TCP protocol. Server must been run before any client run, system will first check if there have enough information to start server (server port and block duration) or client (server IP, server port and client udp port).

Once the server started, server will start to generate tempID for all users and store these in a file (file name as "tempID.txt"), it will repeat this every 15 mins until server been shut down. Server will also create a new hash table to record user unsuccessful login attempt and create multiple threads to communicate with multiple clients.

Once the client start, client will first ask user to make a choice to login or register, if user choose to register, client will record the type-in information and send it to server, server will store this new set of username and password into the credentials file (file name as "credentials.txt"), user will be asked to re-login with these new set of information after register, server will generate tempID for this new user in next generation cycle. If user choose to login, client will record the login information and send it to server, server will check this set of information with information in credentials file to do the authentication checking and send welcome message to client

if username and password are correct or login fail message to client if they are incorrect. If login fail, client will ask user to do the above process again, server will add one unsuccessful attempt to hash table, if user do three unsuccessful logins, server will block user for 60 secs and send a block message to client. Once the client received the block message from server, client will shut down, server will set user unsuccessful attempt to zero after block duration has passed.

Once user successful login, server will start to receive and process command from client, client will ask user to type in the command, if user type some unknow command, client will print error message and ask user to re-type. If user request download tempID, client will send this command to server and server will scan the tempIDs file to find the ID that match the userID and within the time range, then server will send this ID to the client and client will print out on the interface, if this user is new to the server as tempID for this user has not been generated yet, server will send a message to client and tell user to try again later. If user request upload contact log, client will scan the contact log file and send whole information in it to server with the request command, server will read and store these information to a list and start contact log checking, server will scan the tempIDs file and do the comparation, then print out the matched userID with time range, at the end server will send successful message to client to notify the user. If user request logout, client will send this message to server, server will close this thread and client will shut down.

Potential Problems and Improvements

There is very rare chance that server terminal will print some null pointer exception for unknow reasons, but whole features works perfectly. For improvements, it may need to get more time to implement P2P features, and also may need some application layer protocol to encode the message which are transmitting between client and server.