Objective : To develop One to One Android Chat Application

Implementation :

- The android applications act as clients.
- My laptop acts as the chat server.
- Both the android device and the server have to be in the same network.

1. First we have a socket server running. When the android app connects to socket server, the server opens a **TCP connection between server and client**. The server is capable of opening concurrent connections when there are multiple clients. The server also maintains a list of users who are connected.
2. Server accepts an object which consists of details of *from-user*, *to-user*, and the *chat data*. Server forwards this object to the to-user in the object.
3. The client application sends these chat objects to the server and receives the object from the server.
4. Client first connects to the server and selects destination user from a list of names and enters the message to be sent. An object is created with the application-user as the *from-user*, name selected in the list as the *to-user* and message as the *chat data*. This object is passed to the server which in turn forwards to the destination user.
   The destination user receives the object and displays the chat data in its chat window.

Execution Steps :

1. Install Android Studio and Eclipse Oxygen.
2. For running the android application(client), import the project into android studio.
   Install android virtual device and run the application.
   Or connect an android device using USB, enable the Developer Options and run the application.(Installs APK into the android device).
3. For running the Server, import the project into eclipse oxygen and run it.
   Or compile the JavaCmdChatServer.java file and execute it.

Further Improvements that can be done and other points:

1. Server IP address and Port are fixed. (Because in real time they have to be.)
2. Server just displays the logs. But we can backup the logs into a file for further uses.
3. Can dynamically load the users list who are ‘online’ to a client.

4. Need to handle more application crashes by handling all exceptional cases.
5. More pleasing and meaningful UI is needed for better use of the app.
6. Security and encryption of objects being transferred between client and server.

Key Topics :

1. TCP Communication
2. ServerSocket, Socket
3. ObjectInputStream, ObjectOutputStream
4. Data Marshalling and Unmarshalling (especially objects)
5. Seriliazation
6. Threads
7. Android API
8. Andoird Front End Development

Tools:

1. Android Studio 2.3.3

   Android API -24

   Runs on android devices with IcecreamSandwich and above.
2. Eclipse Oxygen.1A

References :

1. https://developers.google.com/android
2. StackOverflow
3. Distributed Computing: Principles and Applications by M.L. Liu
4. The Complete Java Reference by Herbert Schild
5. Professional Android Development by Reto Meier