Homework 5, Distributed Applications Using Android

Network Programming, ID1212

1 Goal

- You can design, develop and deploy an Android app using the Android SDK.
- You can use Android SDK and IDE development tools.

2 Grading

The grading is as follows:

0 points Not passed

1 point Passed, but **no points** for higher grades

2 points Passed, and **one point** for higher grades because an accepted solution was submitted before deadline.

3 points Passed, and **two points** for higher grades because an accepted solution including the optional higher grade task was submitted before deadline.

3 Auto-Generated Code and Copying

You must be able to explain and motivate every single part of your code. You are *not* allowed to copy entire files or classes from the example programs on the course web, even if you understand it and/or change it. However, you are allowed to write code which is very similar to the example programs on the course web. You are also allowed to use GUI builders and other tools that generate code.

4 Task, A Distributed Application Using The Android SDK

You shall develop a distributed application using the Android SDK, you may choose any scenario for the application. A suggestion is to choose one of the problems from homework one, for example the Hangman game.

Requirements on Your Program

All of the following requirements must be met in order for your solution to be accepted.

- Your solution must have an acceptable layered architecture and be well designed. This means it must follow the guidelines of the lecture on architecture, and of the programming examples on the course web. Your are, however, not required to use exactly the same layers as in those examples.
- Some part of the application (e.g. clients) must run on Android devices or emulators of such devices. Some part (e.g. a server) may, but is not required to, execute on a computer connected to the Internet. There must be some network communication from/to the nodes of the application running on Android devices.
- The Android app must have a responsive user interface, which means it must be multithreaded. The user must be able to give commands, for example to quit the program, even if the client is waiting for a message from the server.
- The Android app must have a GUI for user interaction developed using UI Views in Android.
- The server, if there is a server, must be able to handle multiple clients playing concurrently, which means it must be multithreaded.
- The user interface must be informative. The current state of the program must be clear to the user, and the user must understand what to do next.

What is NOT Required of Your Program

Below is an explanation of things that do not affect your score.

- There is no requirement on the complexity of the GUI.
- You are free to choose how the Android app communicates, a suggestion is to use TCP sockets as in homework one.

5 Optional Higher Grade Task

Use Firebase Cloud Messaging, FCM, to communicate with your Android app. You can read about FCM at for example `https://firebase.google.com/docs/cloud-messaging/`. You are free to choose how to use FCM, a suggestion for the hangman game is to send a notification to all apps suggesting them to participate in a "word of the day" contest.

Note that the task is only about handling a FCM message in an Android App. You are not required to send messages programmatically, and certainly not to implement the word of the day contest. It is sufficient to send a notification manually from the firebase console, and make the app display it.