Ranjan Jayapal
CS 522-WS

Report of Submission

In this assignment, you will complete a cloud-based chat app that you started on the previous assignment, where clients exchange chat messages through a Web service. You are given a simple chat server that apps will communicate with via HTTP. You should use the HttpURLConnection class to implement your Web service client, following the software architecture described in class. Set your minimum SDK version to Lollipop (Android 5.1, API 22) for your submission for this assignment. You should ensure that your app works on that platform, although you may develop on another device, e.g., your personal telephone.

1. The main user interface for your app presents a screen with a text box for entering a message to be sent, and a "send" button. The rest of the screen is a list view showing messages received and their senders. You should also have a settings screen where the server URI, and this chat instance's client name and UUID, can be viewed. As with the previous assignment, the chat name and client ID should be set during registration. There should also be, accessible from the options menu, a screen to see all other chat clients.
2. The interface to the cloud service, for the frontend activities, should be a service helper class. This is a POJO (plain old Java object) that is created by an activity upon its own creation, and encapsulates the logic for performing Web service calls. For now, this helper class supports two operations:

- Register with the cloud chat service. This operation will only succeed if the requested client name has not already been registered by a different client.
- Send a message to the cloud chat service.

3. In this assignment, the registration activity runs if the user is not already registered for example, running the application for the first time.
4. After the registration is done, the Chat Activity continues and displays a layout which allows user to enter a message and the name of the chat room the message is intended to.
5. The output, like confirmation of registration and message been received is displayed on the server terminal which is running.
6. The messages sent are displayed on the chat window in the list view which also displays the username the message is from.
7. There is also a server uri field in the registration which allows us to forward our rest requests to that uri. The settings field also shows the user name and client id of the user.
8. In this assignment, the client persists the messages to content provider and later synchronizes the unsent messages or the messages sent by client recently with the server. The synchronization request and response class is used. When a message is sent, the server uploads the message to the client window, deletes the peers or updates them.
9. The synchronization is done every 60 seconds. This constraint cannot be reduced due to the restriction set by android. Although the objective was to update or synchronize the app every random number of seconds between 0 and 10.

Ranjan Jayapal
CS 522-WS

Note- The zip archive “Ranjan_Jayapal” contains 1 zip archive of the android project and 3 files which are APK file, chat application video and README.pdf