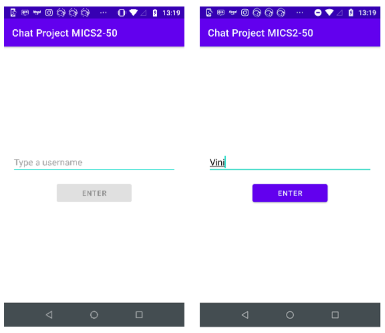
Principles of Software Development - App Report

1 - Code Repository

All the content referenced in this report can be found at **my github page**.

2 - Already implemented

2.1 - Main Activity:

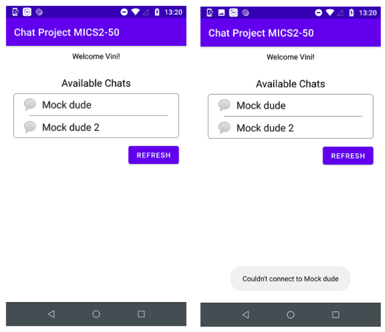


Initial screen of the application, where the user choose its username before connecting to peers. It has 2 main components.

The Username EditText view enables the Enter button when something is typed (or disable if string has length 0). It uses local storage to keep the chosen username and automatically set it on start up. When enter is typed on the keyboard, it is the same as clicking the Enter button.

The Enter Button view creates an Intent to go to Lobby Activity and pass username as extra data. It also saves the current username on local storage.

2.2 - Lobby Activity:

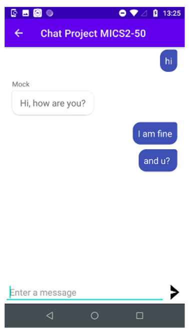


Screen where the user can see and choose a peer to connect. It shows the user the current chosen username by using the passed extra from Main Activity. It registers the broadcast receiver on resume and unregisters on pause. It has 2 main components.

Peers List ListView view updates its content based on the received peers list from the discovery process of Wifi Direct. It uses a drawable for the external border and for the items divisor. Uses layout fragment_peer as its ListItem.

Peer ListItem is a LinearLayout that contains a TextView with a chat bubble Icon on the left of the peer name. When it is clicked it will try to open a connection between the two users (connection not yet implemented, just the call to controller). If the connection can't be done, a toast pops on the bottom of the user screen with an error message.

2.3 - Chat Activity:



hi

Mock

Hi, how are you?

I am fine

and u?

Chat ListView view shows all the messages sent by user and peer (type displayed is controlled by MessageAdapter). Message ListItem represent each message and uses the model Message to pass data into the MessageAdapter. This Message model contains 3 fields: text (the message text), username (the name of who sent the message) and fromUser (True if message was sent by the user, False if it was sent by peer).

MessageAdapter adapter contains the logic to add the message ListItem to the ListView. If is a message from the user, it uses the layout chat_message_user (has only messageBody field), else it uses chat_message_received (has messageBody field and name field).

2.4 - WiFi Direct:

WifiDirectController controls all the necessary logic of WiFi Direct implementation. It implements ConnectionInfoListener to receive the result of setting a connection with a peer (not yet done). It sets the intent filters to be listened to by the application. Also is responsible for discovering peers and updating the list of peers used by Lobby activity. Control the un/register of the BroadcastReceiver.

WifiDirectBroadcastReceiver listen to actions defined on intent filters by WifiDirectController. It Will request peers to WiFi Direct when something is changed in WiFi Direct list or device p2p state (which will be answered on WifiDirectPeersListListener)

WifiDirectPeersListListener just listens to the result of the call for peers list from WiFi Direct made by WifiDirectBroadcastReceiver.

3 - Still to be implemented

- Add connection between user and desired peer (probably by socket communication but still don't know how).
- Add waiting response toast when trying to open a channel with a peer.
- Add toast when someone is trying to connect to user.
- Add transition from lobby to chat activity when connection is established.
- Send message using the connection (currently only mocked messages are sent locally).
- Decide if remove or keep user on peers list after connection to someone.
- Close old connection when connection to other peer is accepted. Or block invites.
- Extra: Add possibility to send files, not just text messages.

4 - Difficulties

The most troublesome part was that i didn't already know android, so i had a long curve of knowledge to be able to do what is already done in the application (it was not easy at all). I believe that if instead of this big application we had smaller simpler ones would be best to learn the content of the lectures.

After That the most troublesome problem was that p2p and WiFi connections are not trivial to test using the emulator of android studio, so it is simpler to test with physical androids if one doesn't know how to set up the environment to use the emulators with WiFi communication. The problem is that in my house there is only 2 androids (an old cellphone and a portable emulator), so i had to set both of them up to test the discovery of peers by the application.

Last problem is exactly understand and implement the communication process between the two applications and manage it without a separated server like for example fire-base (which would help with understanding of API calls, as well as all the communication and connection between clients). It still is a problem for me to figure out a way to build a communication between the users and managing locally the broadcast of messages.