**Linking chat to services.**

After Ending the development of the chat, we found that it’s better to link chat to a specific service, and not allowing users to talk with each other without linking this chat to a certain service, as the core of the app is about services, not social networking with other users.

This approach is also followed by the largest freelancing companies such that Upwork and Freelancer.

**How this looks like in the app?**

Whenever a user requests a service, and other users apply to this service, the asker has two options as in the image below:

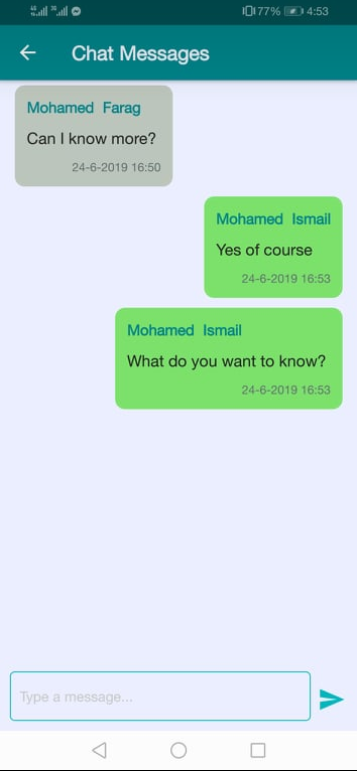


Here we see that this is a proposal for a service, where the asker is asking about someone who can train him in Autocad software , and we can find here that there are two options, the first is to accept this proposal, and the other one is “interview”, this option guarantees a way of communication between the asker and the proposal maker, before hiring them.

We also wanted to make this interview a private one, so we didn’t add it as a group of comments, but we preferred to make it a private chat.

Note that the interview option doesn’t appear to the proposal maker, but we let the initiative to be made by the asker.

Whenever the asker clicks on the interview button, A chat between the two of them (the asker and the helper) is constructed, and a push notification is sent to the helper, whenever the asker sends him a message.

In the images above we see that the asker sent a message to the proposal maker, and that the proposal maker responded to him asking more questions.

This will be a very efficient way to test services providers, before accepting them as service helpers.

Once an agreement happens, the asker can click on the accept button, then as the image below indicates, the “interview” button no longer appears.



As we see hare a “chat with your helper” button is added, instead of the interview button, we also realize that the interview button also disappeared from other proposals, as interviews can no longer be proceeded while there’s a helper for the service.

So the chat can be accessed by two ways: the first way is from the service itself, from a button attached to the proposal, but the second way is from the user profile, where the user can see all the chat that he was involved in since he joined the app.

**Each chat in the chat list is displayed in the following format:**

“The name of the first user” & “The name of the second user” in the service “the name of the service”.

**And here’s an example below for such format:**

