**IEMS5722 Group Project Report**

Members: GU Quanye GE Liqi

App: Personage

A brief introduction of the idea:

This idea stems from a guessing game which often played in our lunch time. The quizzer envision a good topic and participants asked questions to narrow the scope until the they get the right answer. In the process of asking questions, participants can get a lot of new knowledge from the questions raised by others. We think it's a very interesting game to spend time, so we take it as our group project topic.

**1. Problem definition**

As mentioned above, the main role of our app is entertainment, Users can spend their spare time and additionally they may learn some new knowledge during the game.

**2 Features / Functions**

1. Login and registration.

Players can register and log on account, the wrong message will prompt, the device can remember the password.

1. Game logic,

one for the quizzer, the two for the participants, a limited number of questions and unlimited number of questions, the system judge answer automatically and re-assigned quizzer at next round.

1. Real-time update room information and in-game information.

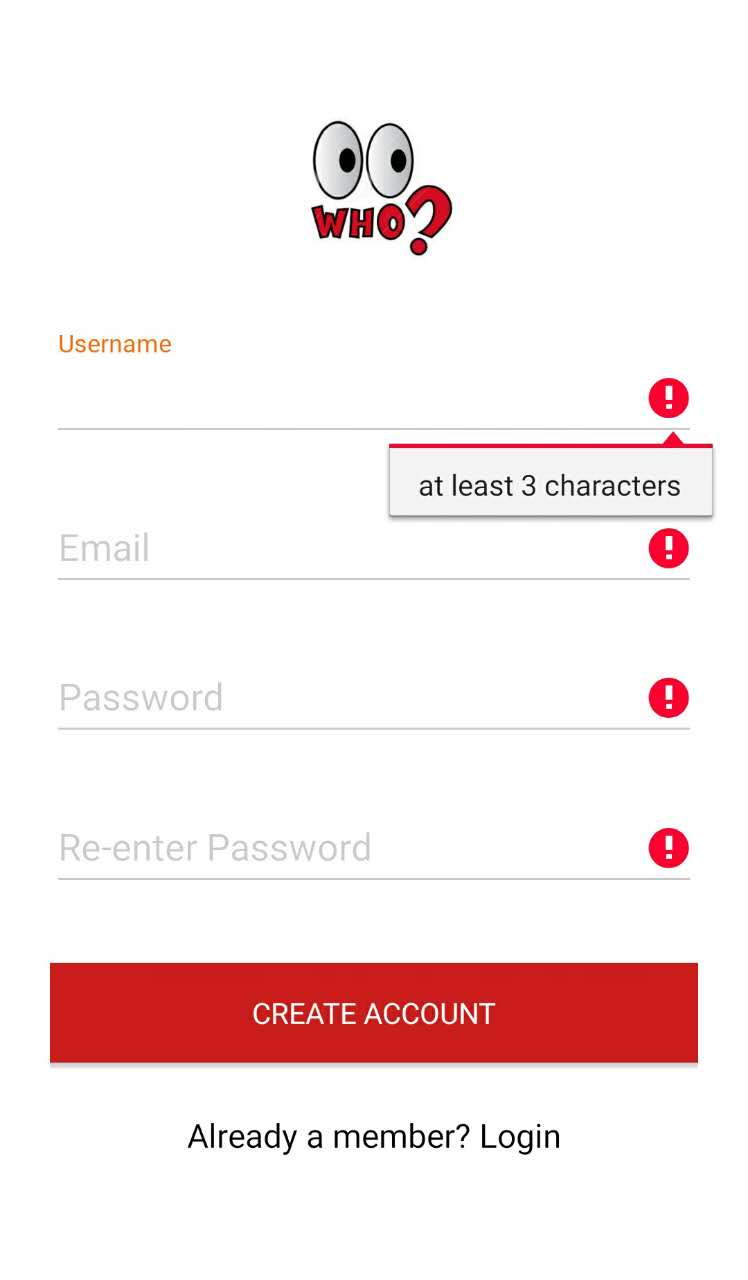
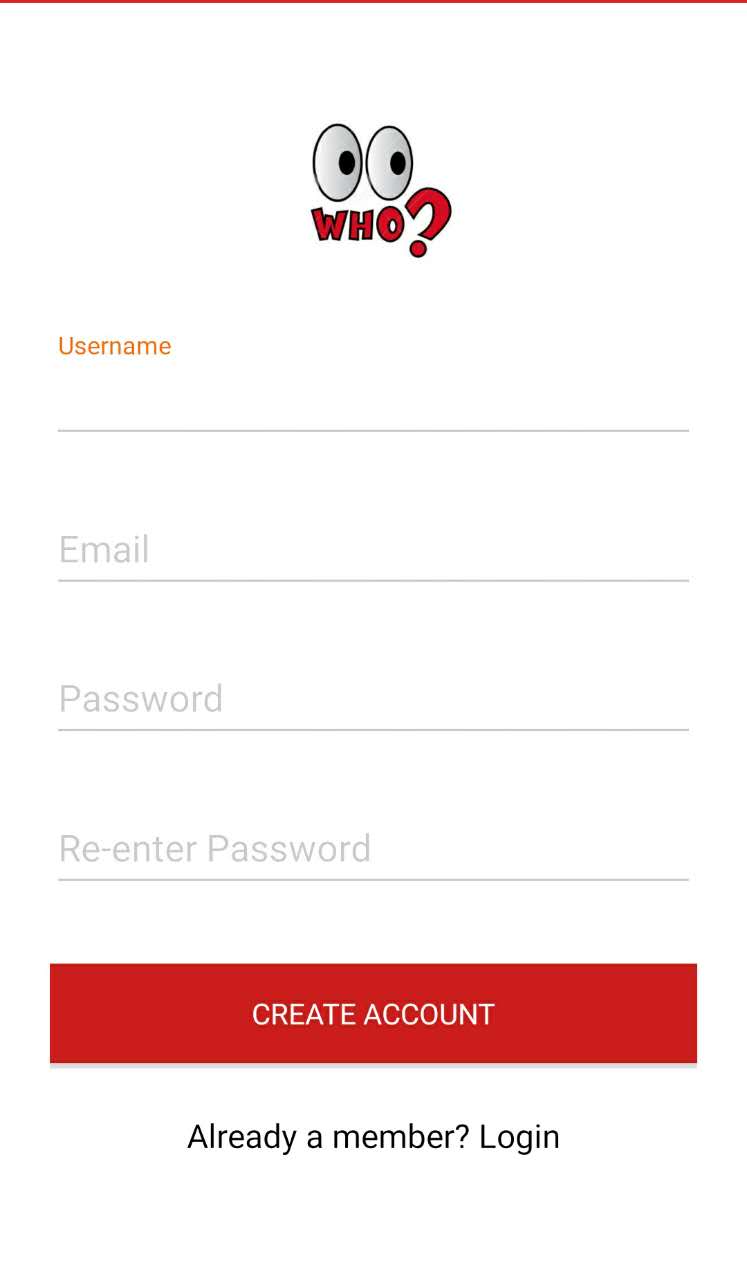
Players in the room list can see the number of rooms in the dynamic change, full of people can not join, enter the room after waiting for the interface, full of people began. In-game questioners will receive a participant's question, and the participant will be able to see the answer given by the questioner.

1. Scoring & rankings.

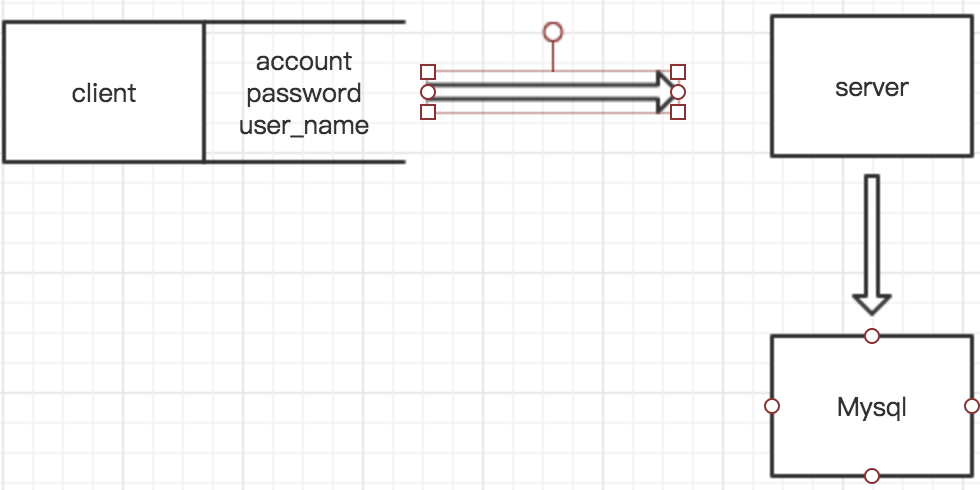
Each round of the game ends with the winner, and after the end of the game, settle and update the rankings and points.

**3. System architecture**

(1). signUp:

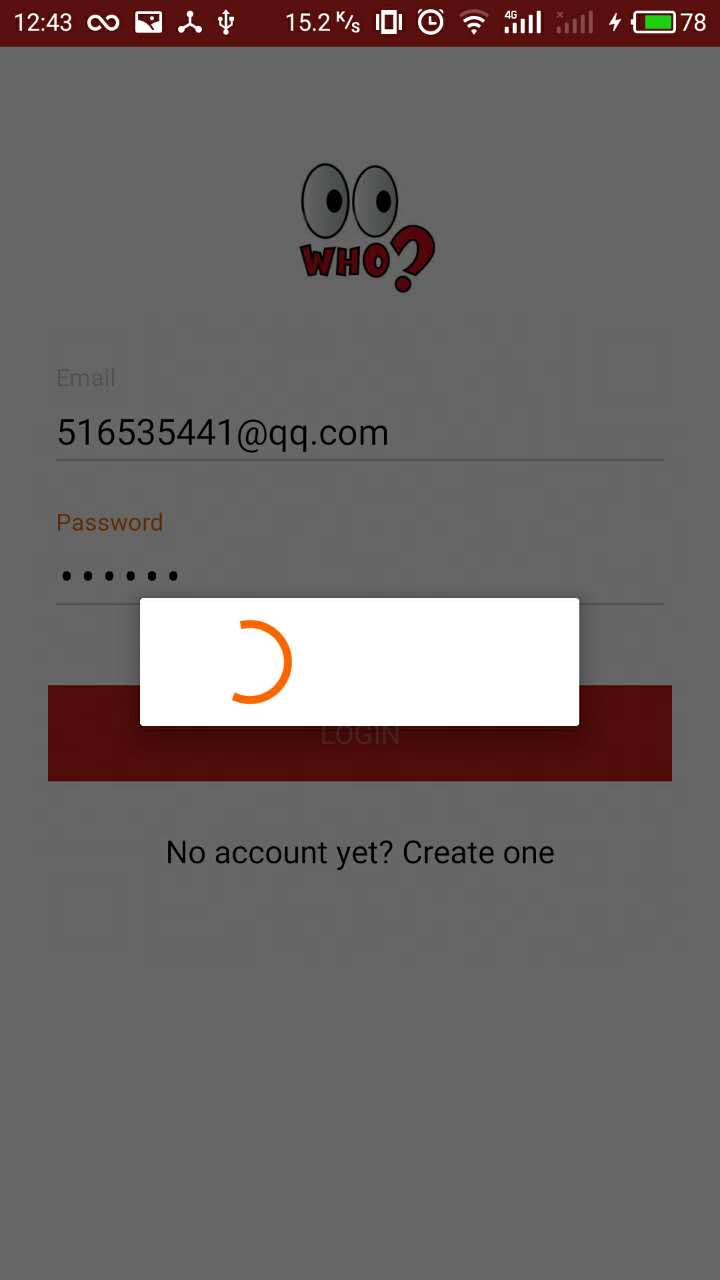
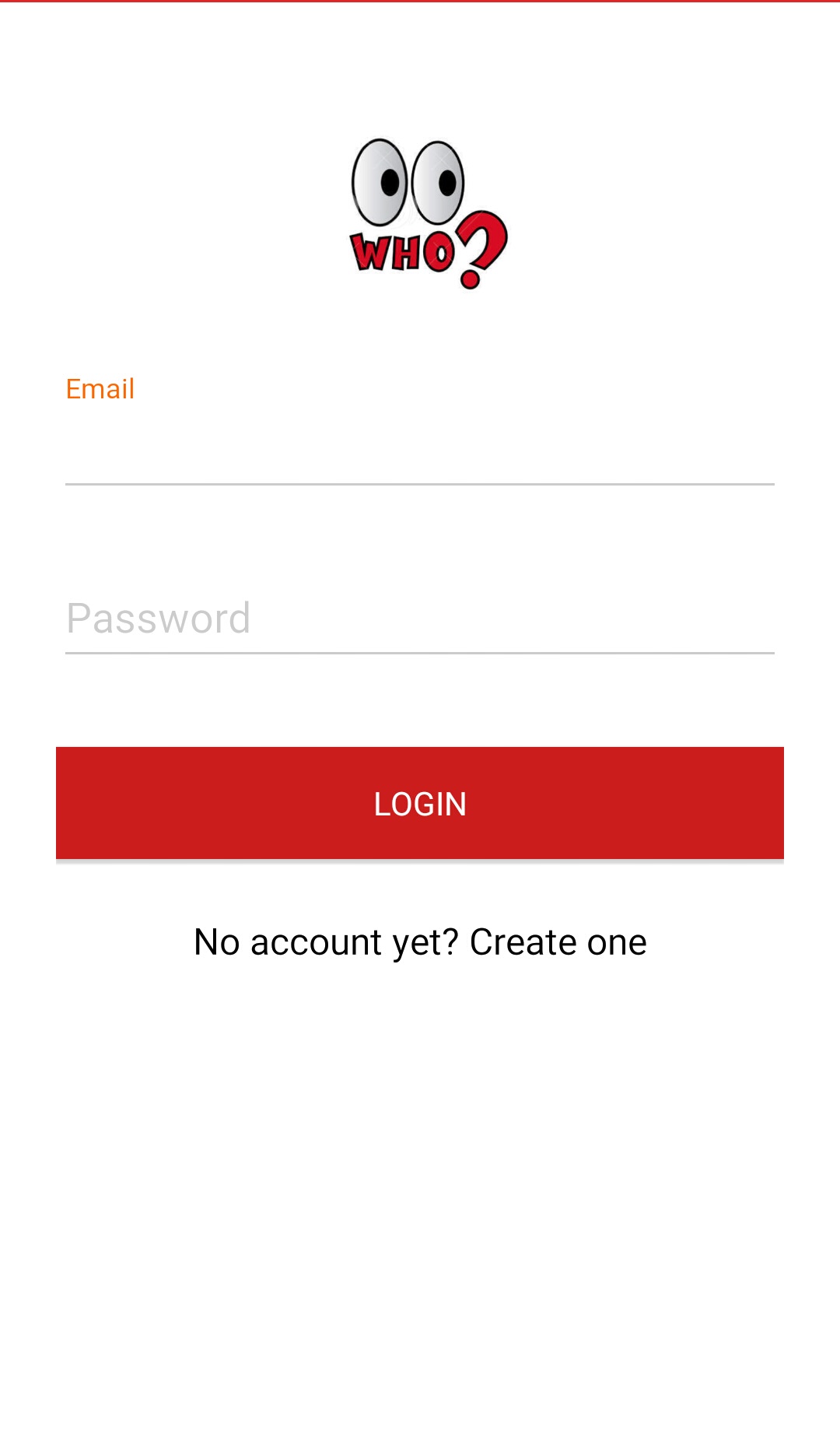


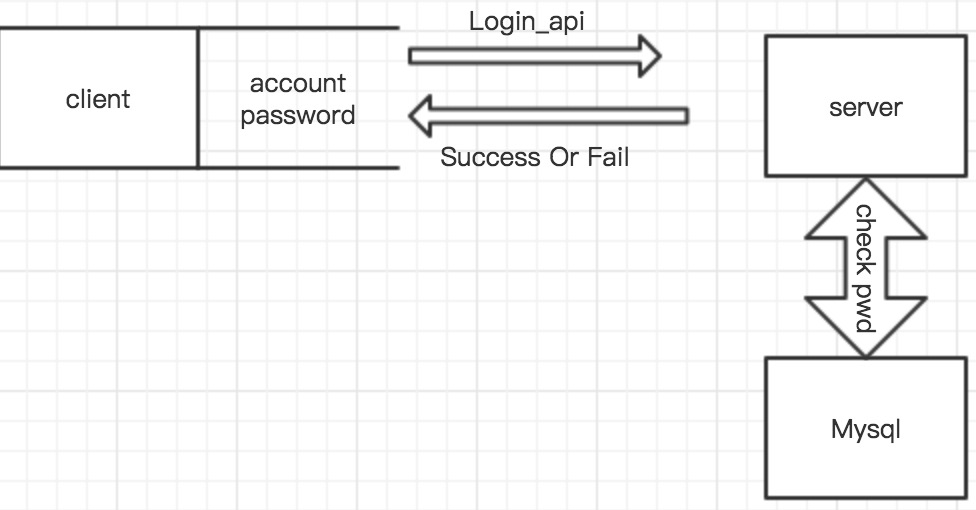
User fill in the EditText and check the format.



Then post the context to the signUp\_api of the server, then server insert the account, name and password into the Mysql users table.

(2). Login

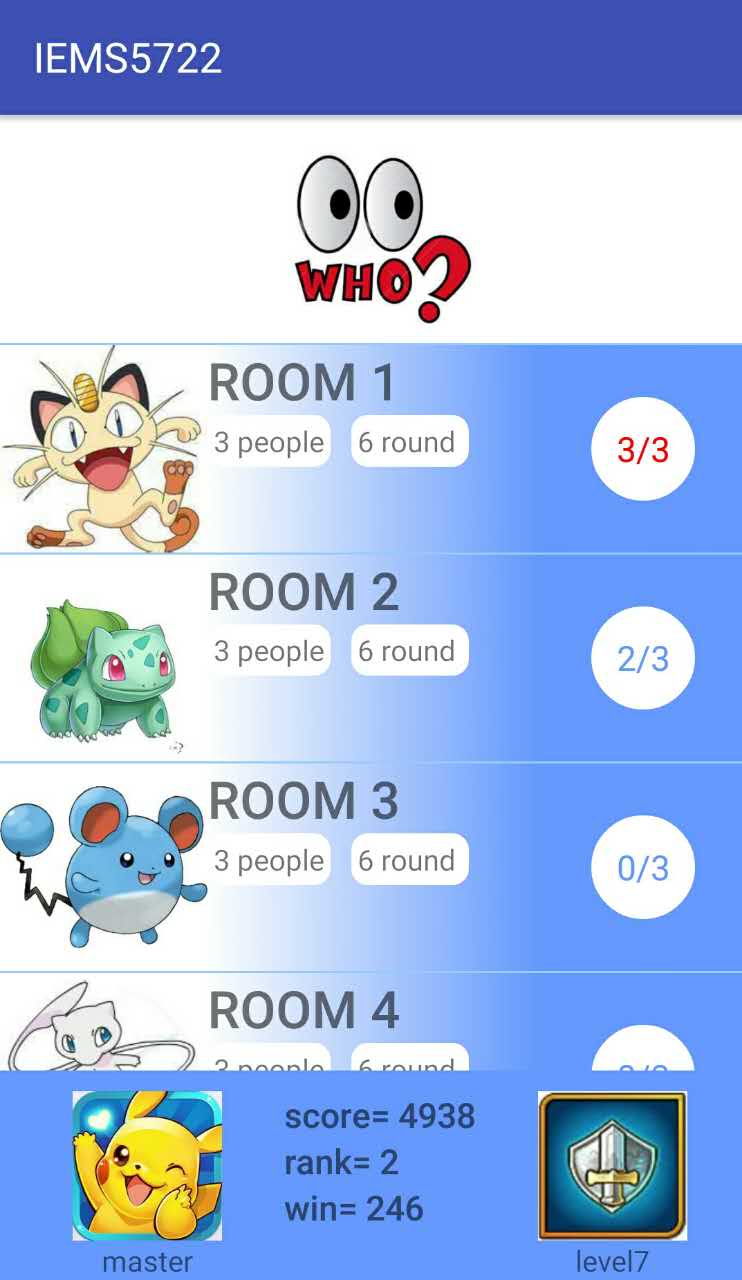
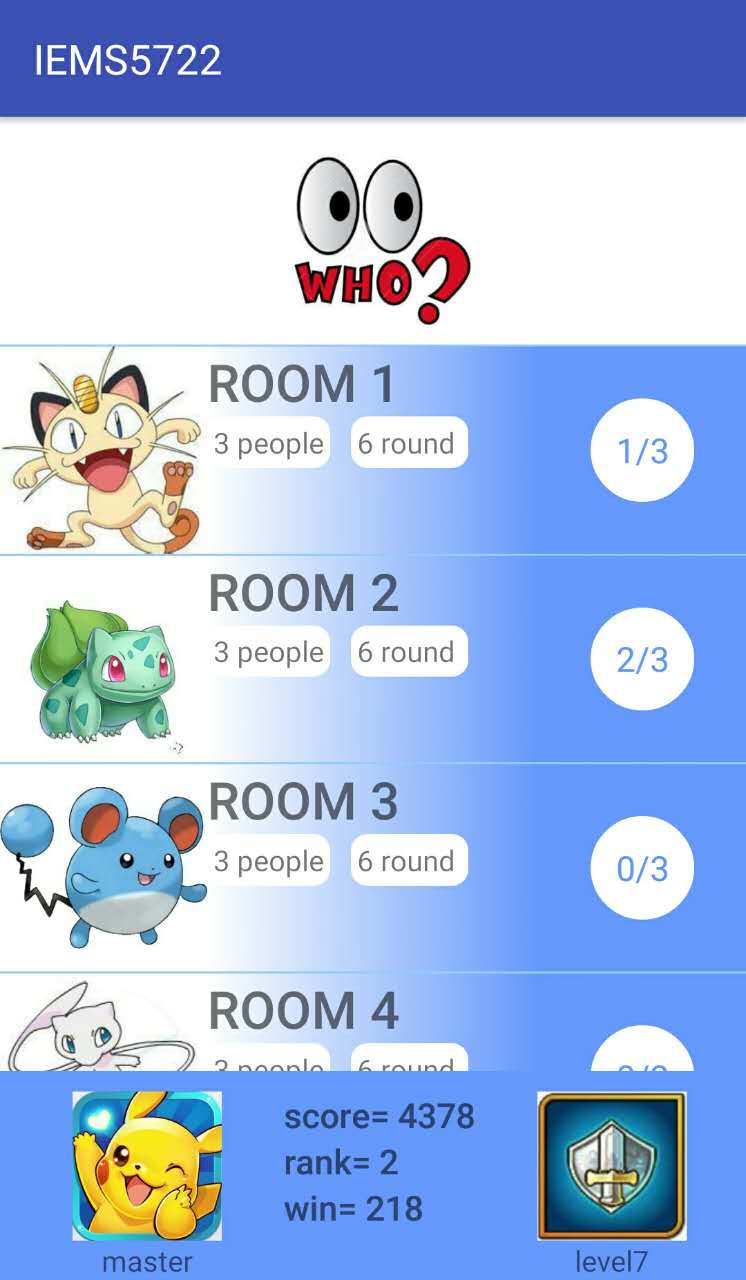
:



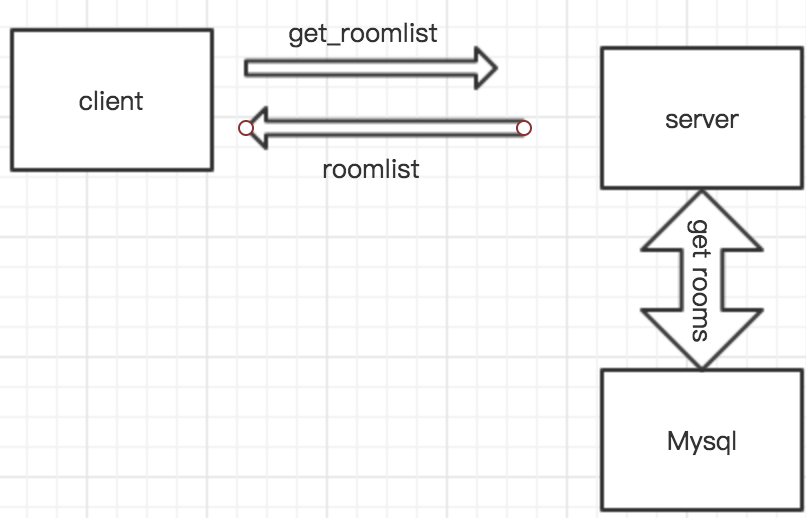
Fill in the EditText and click the login Button, then the client will post the account and password to the server, the server will check whether the database have a user record match the account and password, if not send back a fail code(400), or send a success code(200).

Then if the client get a success code(200), it will login in like the figure 2-2.

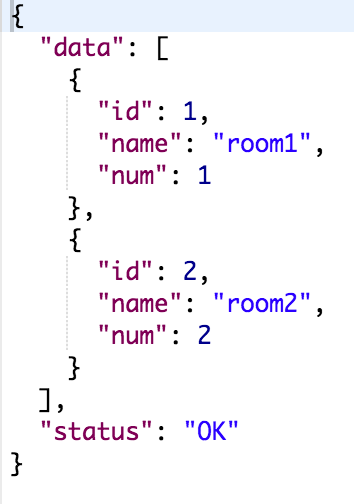
3.RoomList



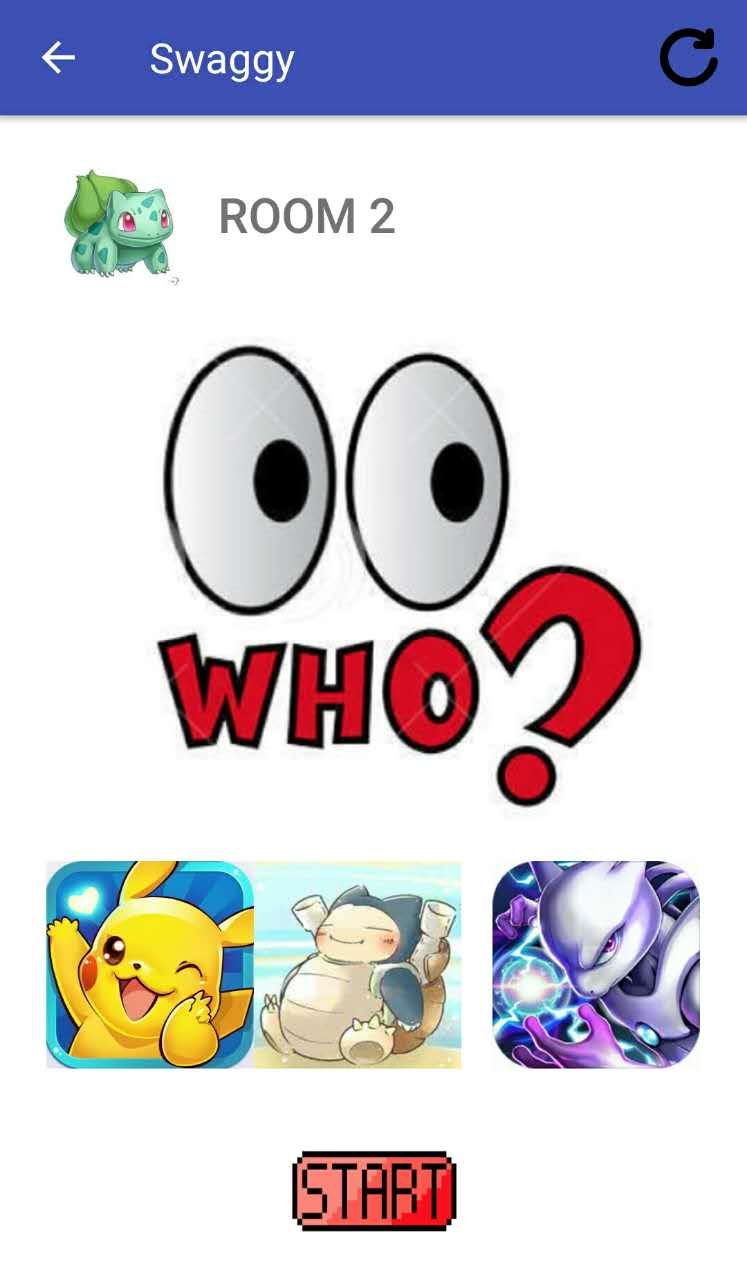
When you after login, you will enter the Roomlist. Show like figure 3-1.



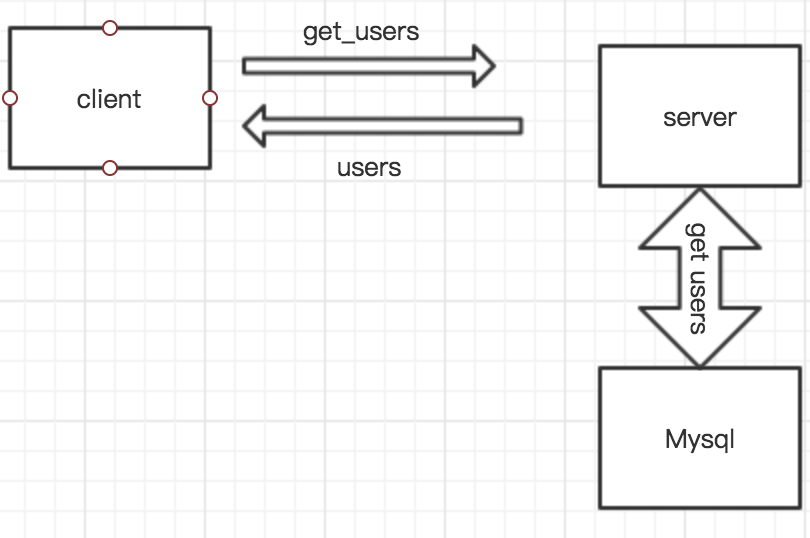
Then the client will get a response like below, show the room which has someone else. And the user can choose a room to enter, and the room which has 3 users already(the maximum users) will not permit the 4th user enter.



(4).Game Room

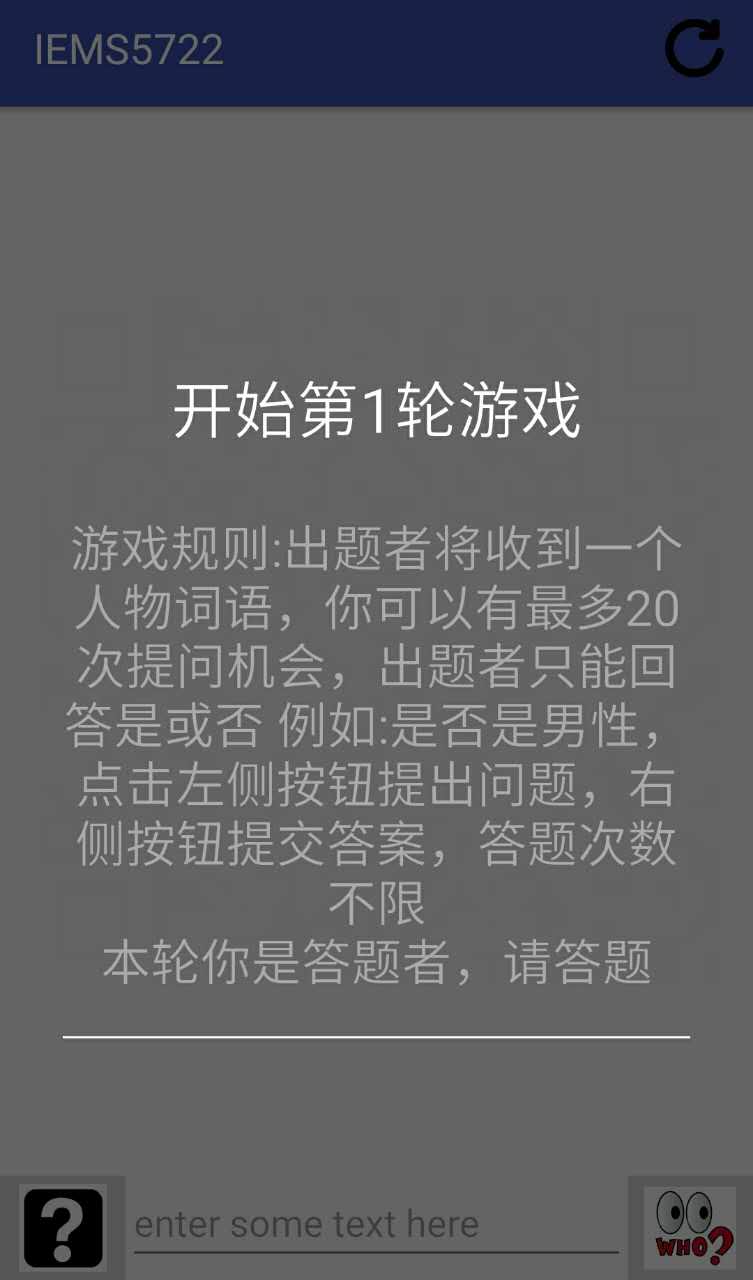
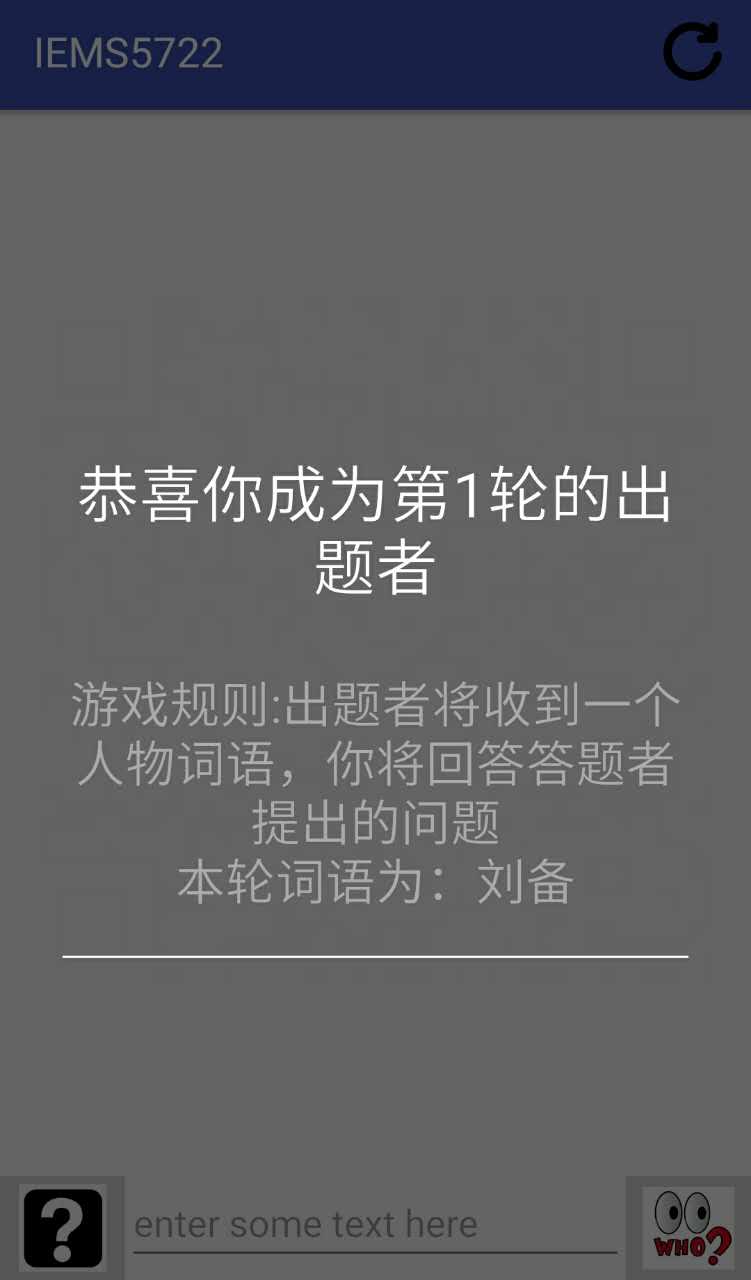


The client request the room’s user from user every 5 second.

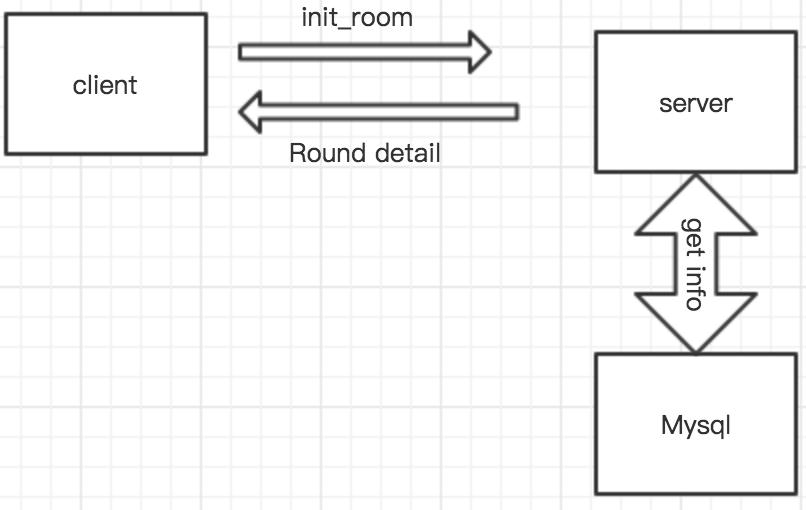


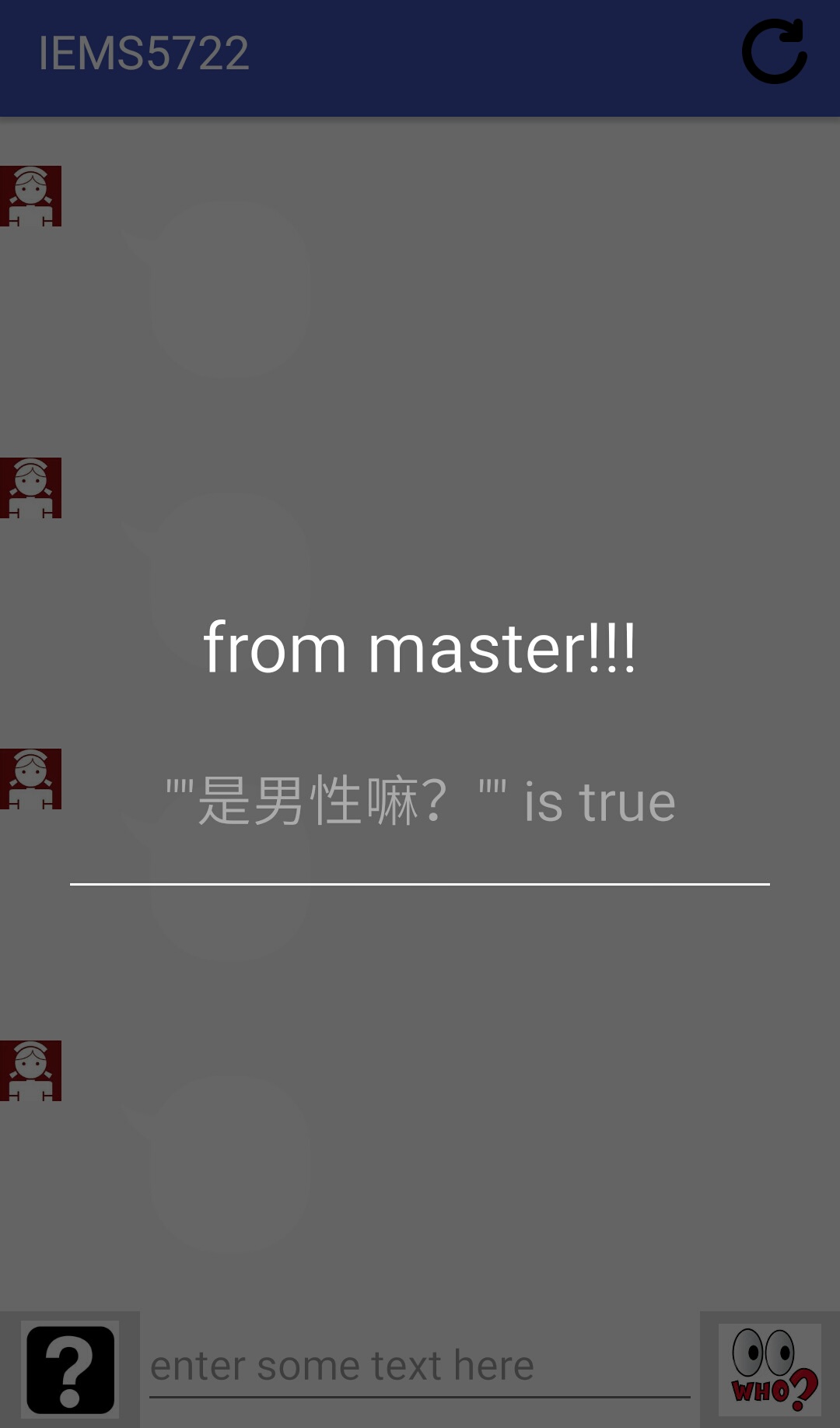
The server will response the exist users in this room, and only when the users number equal 3, the start button will be visible.

(5). Gaming



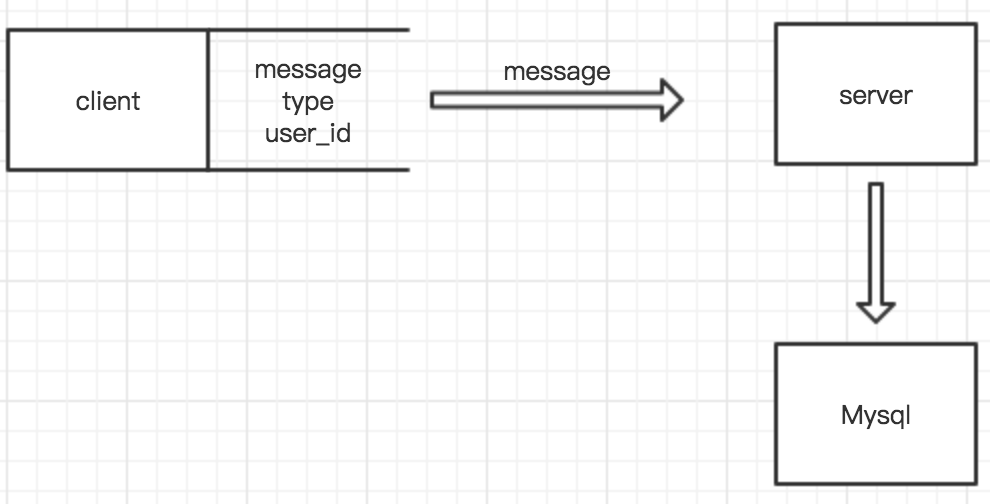
When the game is starting, client will request the server to initiate the room, and the server will response this Round’s master(who know the word and answer the question), and the word.





The user who is not the master can using the “?”button to ask a question to the master, and the master can only answer yes or no, and then the user can get the hint about the right word.

The different is distinguish by the message type set by client. Every message send by user include the question or chat will post to server and insert into database message table.



And user auto get the message from server every 5 seconds.

**4. Further improvement**

(1) players can play through voice.

(2) games increase the timer, the score for winner in each round is related to its spending time.

(3) players can choose their own question.

(4) non-fixed number of players can start the game.

**5. third party libraries**

Okhttp: a library that we use to post and get from server.