Games

A group of friends are all sharing games and tracking them online using a mobile app. The owners will be able to manage their games. The friends will be able to view all the available games, choose one to book and once a game is booked to release it.

On the server side at least the following game details will be maintained:

- Id - the internal game id. Integer value greater than zero.
- Name - the game name. A string of characters representing the game name.
- Type - the game type. Eg. “rts”, “action”, “adventure”, “puzzle”, “shooter”.
- Status - the game status. Eg. “available”, “taken”.
- Size - expressed in GBytes. Also an integer value. Eg. 250, 30, etc.

The application should provide at least the following features:

- (3p) Owner Section (separate activity - available only online)
  a. (1p)**(2p)** The list of his games, at least the name and type should be displayed. The list will be retrieved from the server side using a GET /games call.
  b. (0.5p)**(1p)** Add a game. Using a POST /add call, by sending the game details a game will be added in the list, on success the server will return the game object with the id field set. The name + type fields are identifying a game.
  c. (0.5p) Delete a game. Using POST /remove call, by sending a valid game id, the server will remove the game. On success 200 OK status will be returned.
  d. (1p) Update the game details. Using POST /update call, by sending a valid game object, the server will update the game represented by the game id.
- (3.5p) Friend Section (separate activity - available offline too)
  a. (1p)**(2p)** View the available games. Using GET /unused call. If offline the app will display an offline message and a way to retry the call.
  b. (0.5p)**(1p)** Book a game. The client will be able to reserve a game, if available, using a POST /take call by specifying the game id. The action is available only while online.
  c. **(1p)** Once a friend booked a game, instead of presenting the list of the available games the app will display the details of his booked game, even when offline. Once online the user will be able to release the game. By doing a POST /release with a valid game id. At this point the app will display the available games again.
  d. (0.5p) View the local history with all his played games. For each game that was once played the app will display the game name and the time when the game was played.
  e. (0.5p) Remove the local history.

(1p) On the server side once a new game is added in the system the server will send, using a websocket channel, a message to all the connected applications with the new game object. The application will add the new object in the list of available games.

(0.5p) On all server operations a progress indicator will be displayed.

(0.5p) On all server interactions, If an error message is received, the app should display the error message using a toast or snackbar.

(0.5p) On all interactions, a log message should be recorded.

Only the requirements marked with the **bold** are mandatory, if your laboratory mark is greater or equal to 5.