Bike Sharing App

A group of friends are sharing among them their bikes using a mobile app. Each user is able to view the available bikes in the system and loan and use one at the time. The owner is able to manage his bikes.

On the server side at least the following details are maintained:

- Id - the internal bike id. Integer value greater than zero.
- Name - the bike name. A string of characters representing the bike name.
- Type - the bike type. A string of characters.
- Size - A string of characters representing the bike size.
- Owner - The name of the owner. A string of characters.
- Status - the bike status. Eg. “available”, “taken”, “broken”. A string type.

The application should provide at least the following features:

- User Section (separate activity)
  a. (1p) View the available bikes in a list. Using **GET /bikes** call, the user will retrieve the list of bikes available in the system. If offline, the app will display an offline message and a way to retry the connection and the call. For each bike the name, type and the size are displayed.
  b. (1p) Loan a bike. The user will loan a bike, if available, using a **POST /loan** call, by specifying the bike id. Available online only.
  c. (1p) Once the user loaned a bike, all the bike details are presented to the user.
  d. (1p) Return the bike. The user will have the option to return the bike using **POST /return** call, by specifying the bike id.
  e. (1p) View the historical list of his loaned bikes. The list is persisted on the device, on the local storage, available offline.
- Owner Section (separate activity)
  a. (1p) The list of bikes descending by type. The list will be retrieved using the **GET /all** call, in this list along with the name, type and size, the app will display the status too. Note that from the server you are retrieving an unsorted list.
  b. (1p) Add a bike. Using a **POST /bike** call, by sending the bike object a new bike will be added to the list, on success the server will return the bike object with the id field set.
  c. (1p) Delete a bike. Using **DELETE /bike** call, by sending a valid bike id, the server will remove the bike. On success 200 OK status will be returned.

(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. On all interactions (server or db calls), a log message should be recorded.