

Rent Bikes

A group of friends are all sharing some bikes and they are tracking them using a mobile application. The application has a section for an owner, the person that owns the bike, and a section for a renter, the person who rents a bike. Note that even an owner can play the role of a renter.

Considering the above scenario the application should provide at least the following features:

- Renter Section
 - a. (1p) The list of available bikes, at least the name and the owner of the bike should be displayed. The list will be retrieved from the server side using a GET /bikes call.
 - b. (1p) Rent a bike. Using a POST /rent call, by sending the bike id on success it will return the bike object with the status set to taken.
 - c. (2p) Even when offline, once the user has taken a bike, instead of presenting the list of bikes, the app will display the bike details and a button to release the bike (the bike is marked available again). Once a bike is confirmed to be released (POST /release with bike id), the list of available bikes is displayed once again on the main screen (if online, otherwise it will display a message that we are in the offline mode and offer the choice to retry).
 - Owner Section
 - a. (1p) Available offline. Present details about his/her bike (we assume that an owner has only one bike, no entrepreneurs yet :)). It will display details like:
 - Id - the id of the bike, received from the server once we register a new bike.
 - Name - the name of the bike.
 - Type - the type of the bike. Eg. racing, track, cross, mountain, ...
 - Status
 - Available - if the bike is not taken.
 - Taken - if the bike was rented by an user.
 - b. (2p) By making a POST /register call with at least the following information: name and type, the owner will be allowed to register a bike. On success it will receive a bike object that has at least the id field set. This object will be maintained on the device storage to survive between restarts.
- (1p) On the server side once a bike is registered the server will send, using a websocket channel, a message to all connected applications with the new bike object. The application will add the new object in the list of available bikes. Alternatively the app should check at 5 second intervals if there are new bikes (making the same GET /bikes call) and if it's detecting one it will add the bike on the list.
- (1p) On all server operations a progress indicator will be displayed. Also on all interactions, a log message should be recorded. If error messages are received from the server the app should display them using toast or snackbar messages.