Shopping App

Create a mobile application that allows a family to manage their shopping needs. A family member will be able to:

1. (2p) Retrieve the list of items (GET /items) from the server, while connected. For each item the server will maintain at least the following attributes:
    a. Id - the internal id that identifies the item.
    b. Name - the item name.
    c. Quantity - the item quantity.
    d. Status - the order status:
        i. Pending - the item is not yet purchased.
        ii. Purchased - the item was already purchased.

    In the displayed list, on the main screen, at least the: name, quantity and the status will be presented.
2. (1p) Add a new item by specifying the: name and the quantity. While connected it will send the item to the server by making a POST /add call. As a response the server will return the item with valid: id and status(Pending) fields, if on the server we do not have an item with the exact name, in the pending status. Otherwise (404 Not found) error will be returned.
3. (1p) Buy an item. While connected it will send the item to the server by making a POST /buy call with an object that has at least the id field set. As a response the server will return the same item with the status field set to purchased, if on the server the item was having the pending status. An error (404 Not found) error will be returned otherwise.
4. (1p) While offline the user will be able only to record buy item requests. The buy requests will be maintained on the local storage. Once connected the requests are send to the server.
5. (1p) While offline, on top of the main screen the app will display the number of pending buy requests.
6. (2p) While online when an item is purchased, the app either:
    a. Listens, using a websocket, to new buy messages that are send by the server.
    b. Performs a manual check (using the same GET /orders call). In this case the app will have to see if an item was purchased or not, by comparing the elements received with the ones already in the list of items.

    In either case, the received messages are normal item messages that have a valid id field and the status set to purchased. The received items are updated in the main list (the list of items).
7. (1p) On all server calls, while performing the call, a progress indicator will be displayed. On successful calls (200 OK) an valid item will be returned. On errors (404 Not available or other errors) no item will be returned, the error message will be presented to the user as a toast or snackbar message.