Online Shop App

An online shop is providing his services using a mobile app. The app is used by clients to view, buy or return products and also by the shop employees to manage their goods.

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

- Id - the internal product id. Integer value greater than zero.
- Name - the product name. A string of characters representing the product name.
- Quantity - the number of products available. An integer value greater than zero.
- Type - the product type. Eg. “toy”, “book”, “food”, “dairy”.
- Status - the product status. Eg. “available”, “sold”, etc.

The application should provide at least the following features:

- Client Section (separate activity - available offline too)
  a. (1p) View the offer. Using GET /products call, the client will receive the list of products 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 product the name, quantity and the type are displayed.
  b. (0.5p) Buy a Product. The client will buy a product, if available, using a POST /buy call, by specifying the product id and the quantity. Available online only.
  c. (1p) Once the client bought a product, the list of his products will be displayed. The list is persisted on the device only, in the local storage, available offline too. The client can return a product from the list by doing a POST /return call using the product id, operation available only when online.
  d. (0.5p) View the list of his bought products.
  e. (0.5p) Remove all products from the local list.
- Employee Section (separate activity - available only online)
  a. (1p) The list of all the available products. The list will be retrieved using the GET /all call, here in this list, the app will display the status also.
  b. (0.5p) Add a product. Using a POST /add call, by sending the product object a new product will be added to the list, on success the server will return the product object with the id field set.
  c. (0.5p) Delete a product. Using DELETE /remove call, by sending a valid product id, the server will remove the product. On success 200 OK status will be returned.
  d. (1p) Update the product details. Using POST /update call, by sending a valid product object, the server will update the product represented by the specified id.

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

(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.