Share Items

A group of users are sharing their favorite shopping items using a mobile application. Each user is able to manage their own items.
On the server side at least the following details are maintained:

- Id - the internal item id. Integer value greater than zero.
- Name - the item name. A string of characters representing the item name.
- Description - the item description. A string of characters.
- Image - the item image name. A string of characters representing the image name.
- Category - the item category. A string of characters. Eg. “clothes”, “electronics”, “books”, etc.
- Units - the number of units available. An integer number.
- Price - the item price. A double value.

The application should provide at least the following features:

- Main Section (separate activity)
    A. (1p) View the categories available in the system in a list. Using the **GET /categories** call, the user will retrieve the list of all item categories found in the system. If offline, the app will display an offline message and a way to retry the connection and the call. Once retrieved it should be available offline.
    B. (2p) By selecting a category, the user will be able to get to the list of items that are having that category. To retrieve the list of items having the specified category the **GET /items** call can be used by specifying the category. Once retrieved the list should be available offline.
    C. (1p) Add an item. Using **POST /item** call by specifying all the item details the user will be able to create a new item. Available online only.
    D. (1p) Delete an item. By selecting an item from the list, and using the **DELETE /item** call, the user will be able to delete an item. Available online only.
- Price Section (separate activity)
    A. (1p) The list of top 10 discounted items sorted ascending by price and number of units. The list will be retrieved using the **GET /discounted** call, in this list along with the name, image, and category, the app will display the current price and the number of units. Note that from the server you are retrieving all the items.
    B. (1p) Update the price. From the above list, the user should be able to select an item and increment its price by one unit using the **POST /price** call by specifying the item id.
- (1p) On the server side once a new item is added to the system, the server will send, using a WebSocket channel, a message to all the connected clients/applications with the new item object. Each application, that is connected, will display the received item details, in human form (not JSON text) using an in-app “notification” (like a snackbar or toast or a dialog or a message on the screen).

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