Trading App

Create a mobile application that allows a group of traders to exchange goods on a stock market (the server). The trader (the mobile app) will be able to:

1. (2p) View the available list of goods that are available for purchase (using a GET /goods). For each item from the list it will display:
    a. The name - representing the good name. Eg. Crude oil
    b. The quantity - the available quantity on the market. Eg. 50 barrels.
    c. The price - the price per unit. Eg. 40$ per barrel.
2. (1p) Buy goods. While connected to the stock market, the trader will be able to select an item from the list of goods and by specifying a price and a quantity they will create a buy request to the market (POST /buy). If the market will be able to match the order (have an item with a price lower or equal to the request price and the requested quantity greater or equal) it will respond with an 200 OK, containing the matched price. Eg. if the trader requests 200 tonnes of wheat grains at 100$ per ton, the server will respond with 80$ as the matched price, meaning that it found a better deal. If it will respond with 404 Not Found it means that the market was not able to fulfil the request.
    a. If the request was not matched a message will be printed on the screen and the user will be able to retry (try with a new price and quantity) or cancel (return to the list of good).
    b. If the request was matched the received price will be displayed and the user will be allowed to return main screen (the list of goods).
3. (1p) Sell goods. Even when not connected to the market the trader will be able to add sell requests. To sell a good, the trader will specify the good name, price per unit and quantity.
    a. Once the application is connected the requests will placed on the market, using a POST /sell call. If successful the response (200 OK) will contain the internal id of the good. If the call failed the app will present the error message and allow them to retry or cancel.
    b. While offline the application will maintain the requests in a persistent local storage. Also it will display in the status bar the number of pending requests.
    c. If the sell was successful the application will store the received internal id in a list in memory.
4. (2p) While starting the app, if not connected to the market, the app will present in the status bar that the market is not online and the trader will be able to only add sell requests.
5. (2p) If the market is detecting that one of your goods was bought it will send notifications to all traders, using web sockets.
    a. The message will contain the good id that was sold and the price.
    b. Alternatively the app can offer the trader the choice to manually check if the good was sold (GET /goods call), by providing them with a button to check for status. Here the app will need to establish if the good was sold or not.
    c. If a notification is received for one of the traders sold goods, the app will be display to the trader and the good id will be removed from the memory list.
    d. The list of goods available for purchase will be updated too.
6. (1p) On all market operations, while the app is performing the calls, a progress indicator will be displayed.