School App

Create a mobile application that allows a teacher to manage student attendances. A teacher will be able to:

1. (2p) Retrieve the list of students (GET /students) from the server, while connected. For each student the server will maintain at least the following attributes:
    a. Id - the internal id that identifies the student.
    b. Name - the student full name. Eg. Popescu Vasile.
    c. Group - the student group.
    d. Presences - the total number of presences.

    In the displayed list, on the main screen, at least the: name, group and the current number of presences will be presented.

2. (1p) Add a new student by specifying the: name and the group. While connected it will send the details to the server by making a POST /add call. As a response the server will return the item with valid: id and presences (set to zero) fields, if on the server we do not already have an student with the exact name and group. Otherwise (404 Not found) error will be returned.

3. (1p) Record a presence. While connected it will send the presence to the server by making a POST /presence call with a student object that has at least the id field set. As a response the server will return the same student object with all the attributes and the presences value incremented, only if on the server the student id is found. An error (404 Not found) error will be returned otherwise.

4. (1p) While offline the teacher will be able only to record presence requests. The 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 pending number of presence requests.

6. (2p) While online when an item is purchased, the app either:
    a. Listens, using a websocket, to new messages that are send by the server.
    b. Performs scheduled syncs. Once per 30 seconds the app will refresh the main screen.

    In either case, the received messages are normal student messages that have a valid id field and the number of presences set. The received students 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 student object will be returned. On errors (404 Not available or other errors) no student object will be returned, the error message will be presented to the teacher as a toast or snackbar message.