

Warehouse Inventory Management App

A logistics company is implementing a warehouse inventory management mobile app to track its stock. Warehouse employees can use the app to register new items, query their status, and view various reports.

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

- **Id:** Internal identifier for the item. Integer value greater than zero.
- **Name:** The name of the item. A string of characters.
- **Status:** Current status of the item. A string of characters. Eg. "available", "reserved", "out of stock", etc.
- **Quantity:** An integer value representing the number of items in stock.
- **Category:** The category of the item. A string of characters.
- **Supplier:** The name of the item supplier. A string of characters.
- **Weight:** The weight of the item. A floating-point number.

The application should provide the following features (available without restarting the app):

Inventory Section (Separate Activity/Screen)

A. **(1p)(0.5p) Record an Item:** Record an item using a `POST /item` call by specifying all item details. Available online and offline.

B. **(2p)(1p) View all Items:** View all items in the system using a `GET /items` call. The list should include id, name, category, and status. In offline mode, an offline message and retry option should be provided. The data should persist on the device after retrieval, regardless of online, offline, or restart conditions. Upon successful retrieval, since the data is available on the device, additional server calls should not be performed.

C. **(1p)(1p) View Item Details:** By selecting an item from the list, the user can view all the details. Using `GET /item` call with the item id, the data should be retrieved from the server each time and made available on the device.

D. **(0.5p) Delete an Item:** Delete an item from the list, using a `DELETE /item` call by sending the item ID. The action should be available in the details screen.

Report Section (Separate Activity/Screen) - Available Online Only:

A. **(1p)(0.5p) View Categories:** View all available item categories in a list using a `GET /categories` call.

B. **(1p)(0.5p) View Items by Category:** View all available items from the selected category in a list using the `GET /byCategory` call, by specifying the selected category.

C. **(1p)(0.5p) Top 10 Heaviest Items:** View the top 10 heaviest items in a list with details such as name, status, weight, and category. Use the `GET /all` call and present the result in descending order by weight.

D. **(0.5p) Top 5 Suppliers:** View the top 5 suppliers in a list with the supplier name and the number of items supplied. Use the same `GET /all` call and present the result in descending order by the number of supplied items.

Supplier Section (Separate Activity/Screen) - Available Online Only:

A. (1p) **Record Supplier's Name:** Record the supplier's name in application settings. Persisted to survive app restarts.

B. (1p) **View Supplier's Items:** View all items of the persisted supplier in a list showing item name, status, and quantity. Use `GET /supplier-items` call by specifying the supplier.

Additional Functionalities:

- **(1p)(1p) Server Notifications:** 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 object. Each application that is connected will display only the item name, status and category, using an in-app "notification" (e.g., using a snack bar, toast, or an on-screen dialog).
- **(0.5p)(0.5p) Progress Indicator:** During all server or database operations, a progress indicator will be displayed.
- **(0.5p)(0.5p) Error Handling & Logging:** On all server or DB interactions, if an error message is received, the app should display the error message using a toast or snackbar. A log message should be recorded for all interactions (server or DB calls).

NOTE: If your laboratory grade is at least 4.5, only the **bold** points will be used to compute the exam grade.