Real Estate App

A group of users is looking to buy, sell or rent properties using a mobile application. Each user is able to manage their own property listings.

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

- Id - the internal property-id. Integer value greater than zero.
- Date - the date when the property was listed. A string in the format "YYYY-MM-DD".
- Type - the type of property, such as House, Apartment, Townhouse, etc. A string of characters.
- Address - the physical address of the property. A string of characters.
- Bedrooms - the number of bedrooms in the property. An integer.
- Bathrooms - the number of bathrooms on the property. An integer.
- Price - the price of the property. A double value.
- Area - the square footage of the property. An integer value.
- Notes - any additional notes added by the user. A string of characters.

The application should provide at least the following features:

- Main Section (separate activity)

  A. (1p) View the list of properties. Using the **GET /properties** call, the user will retrieve the list of all the properties and display the address. If offline, the app will display an offline message and a way to retry the connection and the call. Once retrieved, the data should be available, even offline.

  B. (2p) By selecting a property, the user can view the details of that property. To retrieve the details of a specific property, the **GET /property** call can be used by specifying the property-id. Once retrieved, the data should be available, even offline.

  C. (1p) Add property listing. Using **POST /property** call by specifying all the property details, the user will be able to create a new property listing. Available online only.

  D. (1p) Delete property listing. By selecting a property from the list, and using the **DELETE /property** call, the user will be able to delete a property listing. Available online only.

- Search Section (separate activity)

  (2p) View properties based on certain search criteria, such as property type, price, and the number of bedrooms. The list will be retrieved using the **GET /search** call. The list should present the properties descending by date, and ascending by price.

- (1p) On the server side, once a new property is added to the system, the server will send, using a WebSocket channel, a message to all the connected clients/applications with the new property object. Each application, that is connected, will display the received property details, in human form (not JSON text) using an in-app “notification” (like 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 snackbar. A log message should be recorded on all interactions (server or DB calls).