Bookmark App

A group of users are sharing their favorite bookmarks using a mobile application. Each user is able to manage the bookmarks.

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

- Id - the internal bookmark id. Integer value greater than zero.
- Name - the bookmark name. A string of characters representing the bookmark name.
- Description - the bookmark description. An string of characters.
- Url - the bookmark url. An string of characters.
- Type - the bookmark type. A string of characters. Eg. “text”, “video”, “page”, etc.
- Rating - the bookmark rating. An integer value.

The application should provide at least the following features:

- Main Section (separate activity)
  a. (1p) View the types available in the system in a list. Using **GET /types** call, the user will retrieve the list of all bookmark types 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 type, the user will be able to get to the list of bookmarks that are having that type. To retrieve the list of bookmarks having the specified type the **GET /bookmarks** call can be used by specifying the type. Once retrieved the list should be available offline.
  c. (1p) Add a bookmark. Using **POST /bookmark** call by specifying all the bookmark details the user will be able to create a new bookmark. Available online only.
  d. (1p) Delete a bookmark. By specifying the bookmark id, using the **DELETE /bookmark** call, the user will be able to delete a bookmark. Available online only.
- Rate Section (separate activity)
  a. (1p) The list of top 10 underrated bookmarks sorted ascending by rating. The list will be retrieved using the **GET /underrated** call, in this list along with the name, url and type, the app will display the current rating. Note that from the server you are retrieving all the bookmarks.
  b. (1p) Rate a bookmark. From the above list the user should be able to select a bookmark and increment its rating by one unit using the **POST /rate** call by specifying the bookmark id.

(1p) On the server side once a new bookmark is added in the system, the server will send, using a websocket channel, a message to all the connected clients/applications with the new bookmark object. Each application, that is connected, will display the received bookmark 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. On all interactions (server or db calls), a log message should be recorded.