

Dinner App

An dinner is sharing his tables online using a mobile app. The owners will be able to manage their tables. The clients will be able to view all the available tables, book a table and release it.

The application should provide at least the following features:

- Client Section (separate activity - available offline too)
 - a. (1p)(**2p**) View the available tables. Using GET /tables call all the free tables. If offline the app will display an offline message and a way to retry the call.
 - b. (0.5p)(**1p**) Book a table. The client will be able to reserve a table, if available, using a POST /reserve call by specifying the table id. Available online only.
 - c. (**1p**) Once a client reserved a table, instead of presenting the list of the available tables the app will display the details of his booked table, even when offline. Once online the user will be able to release the table. By doing a POST /release with his table id. After this, the app will display the available tables again.
 - d. (0.5p) View the local history with all his used tables. For each table that was once used the app will display the table name, the seat number and the time when the table was used.
 - e. (0.5p) Remove the local history.
- Owner Section (separate activity - available only online)
 - a. (1p)(**2p**) The list of his tables, at least the name, seat count and type should be displayed. The list will be retrieved from the server side using a GET /all call.
 - b. (0.5p)(**1p**) Add a table. Using a POST /new call, by sending the table details a table will be added to the list, on success the server will return the table object with the id field set. The name + type fields are identifying a table.
 - c. (0.5p) Delete a table. Using POST /delete call, by sending a valid table id, the server will remove the table. On success 200 OK status will be returned.
 - d. (1p) Update the table details. Using POST /update call, by sending a valid table object, the server will update the table represented by the table id.

(**1p**) On the server side once a new table is added in the system the server will send, using a websocket channel, a message to all the connected applications with the new table object. The application will add the new table in the list of available tables.

(**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.

(0.5p) On all interactions, a log message should be recorded.

On the server side at least the following table details will be maintained:

- Id - the internal table id. Integer value greater than zero.
- Name - the table name. A string of characters representing the table name.
- Seats - the number of seats. An integer value greater than zero.
- Type - the table type. Eg. "small", "big", "private".
- Status - the seat status. Eg. "free", "used".

Only the requirements marked with the **bold** are mandatory, if your laboratory mark is greater or equal to 5.