

Theater App

A cinema is managing their reservations using a mobile application. The users are able to view all the available seats and make reservations. Once a seat is confirmed to be reserved they are able to pay for their ticket.

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

- Id - the internal seat id. Integer value greater than zero.
- Name - the seat name. A string of characters representing the seat name. I.e. #12
- Type - the seat type. Eg. "stall", "royal circle", "grand circle", "balcony". A string type.
- Status - the seat status. Eg. "available", "reserved", "confirmed", "taken". A string type.

The application should provide at least the following features:

- Client Section (separate activity - available offline too)
 - a. (1p) View the available seats. Using **GET /seats** call, the user will receive the list of all the available seats in the system. If offline the app will display an offline message and a way to retry the connection and the call. For each seat the name and the type are displayed.
 - b. (0.5p) Reserve a seat. The user will be able to reserve a seat, using a **POST /reserve** call, by specifying the seat id. Available only when online. The server, on success, will return the seat with the status changed to "reserved".
 - c. (0.5p) Once a seat is reserved, the application in the client section, will display, even if offline, all the details related to the reserved seat. Using an refresh button the user will be able to call **GET /refresh** by specifying a seat id to retrieve the status of the seat. The received information is persisted locally.
 - d. (0.5p) Once the status of the seat is "confirmed" the user will be able to buy the seat using a **POST /buy** call, by specifying the seat id. On success the server will return the seat with the status set to "taken". Available only if online.
 - e. (0.5p) Delete the local stored seat details.
 - Theater Section (separate activity - available only online)
 - a. (1p) The list of all the seats. The list will be retrieved using the **GET /all** call, in this list along with the name, hour and type, the app will display the status also.
 - b. (1p) Confirm a reserved seat. Using **POST /confirm** call, by specifying a seat id, the seat status will be changed to "confirmed".
 - c. (0.5p) Mark all seats as available, using **DELETE /clean** call.
 - Admin Section (separate activity - available only online)
 - a. (0.5p) List all the confirmed seats, using **GET /confirmed** call.
 - b. (0.5p) List all the purchased seats, using **GET /taken** call.
 - c. (1p) Delete all the seats. Using **DELETE /zap** call.
- (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 (server or db calls), a log message should be recorded.