

Library Management System

In a library management system, users can access a mobile application to browse and interact with the library's collection. Employees, on the other hand, can manage the library's inventory.

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

- Id - An internal identifier for each book. Integer value greater than zero.
- Title: The title of the book. A string of characters representing the book title.
- Author: The name of the author. A string of characters.
- Genre: The genre of the book. A string of characters.
- Quantity: The number of available copies of the book. An integer value.
- Reserved: The number of copies reserved by users. An integer value.

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

- Client Section (separate activity/screen)

A. (2p) Users can view a list of available books in the system in a list. Using **GET /books** call, the user will retrieve the list of all books in the system. The title, author, and genre should be displayed for each book. Once retrieved, the server call should not be performed again, the data should be available on the device, regardless of whether it is online, offline, or restarts.

B. (1p) By selecting a book from the list, users can reserve a book by specifying the book id using **PUT /reserve** call. Available only while online.

C. (1p) List reserved books. Using **GET /reserved** call the user should be able to view the list of reserved books. Once retrieved, the server call should not be performed again, the data should be available on the device, regardless of whether it is online, offline, or restarts.

D. (1p) Borrow a book. Users can borrow a book by using **PUT /borrow** call and specifying the book id of the selected book from the reserved book list. Available only while online.

- Employee Section (separate activity/screen) - Available online only.

A. (1p) Employees can view the list of all books available in the system, sorted by genre and quantity. The list should be presented in ascending order by quantity. Using the same **GET /books** call. The call should be performed each time when the employee enters this section.

B. (1p) Employees can add a new book to the system by using **POST /book** call and sending all the book details.

- (1p) On the server side, once a new book is added to the system, the server will send, using a WebSocket channel, a message to all the connected clients/applications with the new book object. Each application that is connected will display the received book details, in a human form (not JSON text or toString) using an in-app "notification" (like a snack bar or toast or a dialog on the screen).

(0.5p) On all server or DB operations, a progress indicator will be displayed.

(0.5p) 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 on all interactions (server or DB calls).