

Chat App

A group of friends are able to use a mobile application to exchange messages.

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

- Id - the internal message id. Integer value greater than zero.
- Sender - the user that is sending the message. A string of characters.
- Receiver - the user that should receive the message. A string of characters.
- Text - the message body. A string of up to 150 characters.
- Date - the date of creation. An integer value representing the unix time.
- Type - the type of the message. A string of characters. Eg. "public", "private".

The application should provide at least the following features:

- Public Section (separate activity - available online only)
 - (1p) View the latest 10 messages retrieved from the server. Using **GET /public** call all the public messages are retrieved from the server. Note that you should display only the latest 10 ordered descending by the date field. If offline the user should have the option to retry the operation.
 - (1p) View all the users in a list, in an ascending order. To retrieve the list of users you should use the **GET /users** call.
 - (1p) When selecting a user from the above list, show all the public messages that are having him as a sender.. Using **GET /sender** call by specifying the user name you will get all such messages. The list should be sorted descending by date.
 - (1p) When selecting a user from the above list, show all the public messages that are having him as a receiver.. Using **GET /receiver** call by specifying the user name you will get all such messages. The list should be sorted descending by date.
 - (1p) Post a message. Using **POST /message** call by specifying all the message fields (except id and date) a new message will be created. Note that the user should be allowed to send both private and public messages from this section.
 - Private Section (separate activity)
 - (1p) Give the user the option to store his chat username in the local storage. This operation should be done only once.
 - (1p) Retrieve all his private messages. Using **GET /private** call, by specifying the username. The list should be sorted descending by date and available offline too.
 - (1p) Delete message. From the above list the user should be able to delete a message by using the **DELETE /message** call using the message id. Available online only.
- (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.