



New Emoji Picker for Mobile Client

Google Summer of Code 2022

Basic information

Name	Danish Ahmed Mirza
GitHub	try-catch-stack
email	danishmirza30602@gmail.com
Nationality	Indian
University	Indian Institute of Technology (BHU) , Varanasi
Degree	Bachelor of Technology
Timezone	IST (UTC +5:30)

About Me

I'm **Danish** , a second year engineering student at the **Indian Institute of Technology (BHU), Varanasi** . I'm an enthusiastic programmer and love to explore new technologies .I have a keen interest in all sorts of software development, currently aligned more towards web and mobile development. The interest and motivation has further increased since I started contributing to open source. I constantly strive to improve my skills and I'm open to learning new technologies.

Prior Experience

I have worked on several frontend and backend projects with varied tech stacks like React.js, Vue.js, Django, React Native, Nodejs and Express to name a few. I started contributing to open source projects almost 6 months ago and have not looked back since then. Its vibrant , supporting and engaging community is something which keeps me motivated .

I got to know about [Rocket.Chat](#) while exploring open source projects around the month of October last year and it was then ,I knew I definitely wanted to contribute to it in any way possible. I am really thankful to the amazing community which helped me to get started.

Project - New Emoji Picker for Mobile Client

Project duration : **350 Hours**

Overview

Action sheets are a specific type of alert that appears in response to a control or action and presents one or multiple options to the user to select. Action sheets are a very common UX pattern used in the apps used for users to interact with quick actions and can be seen in almost all the apps. Some examples of Action sheets used in the Rocket.Chat app are message actions menu, share view etc.

The current implementation of Action sheets in Rocket.Chat mobile clients is based on the library [react-native-scroll-bottom-sheet](#) which is an awesome library but has not been updated for over 2 years and also lacks some features. [react-native-bottom-sheet](#) has grown a lot recently and has been loved by the community. It also provides some of the missing features like extracting scrollable content to allow developers customize the sheet content, like integrating React Navigation as the sheet content. That's why we want to migrate the Action Sheets to it .

We want the Action Sheets to be dynamic enough to render the emoji picker inside of it. Currently , the the behavior for using emojis is divided into these ways:

- Emoji keyboard : It is the component present on the left of the message box and it renders the Emoji picker component in the form of a custom keyboard .
- Reaction picker : It is based on a modal that appears when we want to react to a message. It also renders the same Emoji picker component used in the Emoji keyboard but in the form of a modal.

So the idea is to redesign this Emoji picker component , improving the overall interface and user experience .This Emoji picker will then be used in the Emoji keyboard. And for the reaction picker, it'll be used in the Bottom Action sheet .

Mentors	Diego Mello , Daniel Silva
----------------	--

Implementation

The first goal of the project is to migrate the ActionSheets to [BottomSheetView](#). To start with, I'll install and set up [react-native-bottom-sheet](#) . No other dependencies are to be installed as `react-native-reanimated` and `react-native-gesture-handler` , the dependencies required by the library are already being used in the Rocket.Chat mobile apps.

Once the library is set up ,I would start the implementation of ActionSheets, taking the current designs as inspiration. As we want the ActionSheets to be dynamic enough to render Emoji Picker and other views, we do not want to predefine the components to be used in it and rather pass them as children wherever the Action Sheet is used.

The approach I plan to use is to make a component `<BottomSheetContent/>` which will render the components for the ActionSheets . It will return a `<BottomSheetFlatList/>` or `<BottomSheetView>` with the custom components like the Emoji Picker, depending on where the ActionSheet is used.

The `<BottomSheetContent/>` will be implemented something like this:

```

1  const renderFooter = () =>
2    data?.hasCancel ? (
3      <Button
4        onPress={hide}
5        style={[styles.button, { backgroundColor: themes[theme].auxiliaryBackground }]}
6        theme={theme}
7        accessibilityLabel={I18n.t('Cancel')}>
8        <Text style={[styles.text, { color: themes[theme].bodyText }]}>{I18n.t('Cancel')}</Text>
9      </Button>
10    ) : null;
11
12  const renderItem = ({ item }: { item: IActionSheetItem['item'] }) => <Item item={item} hide={hide} />;
13
14  if (type === 'FlatList') {
15    return (
16      <BottomSheetFlatList
17        data={data.options}
18        refreshing={false}
19        keyExtractor={item => item.title}
20        bounces={true}
21        renderItem={renderItem}
22        style={{ backgroundColor: themes[theme].focusedBackground }}
23        keyboardDismissMode='interactive'
24        indicatorStyle='black'
25        contentContainerStyle={styles.content}
26        ItemSeparatorComponent={List.Separator}
27        ListHeaderComponent={List.Separator}
28        ListFooterComponent={renderFooter}
29      />
30    );
31  }
32  return <>{children}</>;

```

This behavior can be achieved by passing these props while calling the `ShowActionSheet()` function exposed by the `useActionSheet()` hook. The options will have the following type:

```

1  type TActionSheetOptions = {
2    options: TActionSheetOptionsItem[];
3    headerHeight: number;
4    customHeader: React.ReactElement | null;
5    hasCancel?: boolean;
6    type?: string;
7    children?: React.ReactElement | null;
8  };

```

Emoji Picker

The current EmojiPicker component used in the Emoji Keyboard and the Reactions modal uses a `ScrollableTabView` component with the categories as the TabBar and clicking one of these opens a FlatList with some number of columns of emojis depending on the width of the device.

- New Emoji Picker

The new design plans to bring in the search emoji functionality with scrollable section lists for the categories of emojis.

Another handy feature that I plan to implement in the new Emoji picker is the backspace/delete button present on the bottom right corner which will make it easier to delete the emoji from the EmojiKeyboard itself.

- Searching Emojis

Search emojis function can be implemented similar to the function used in the suggestions of the emojis that appear on using the ':' symbol while typing a message.

The search emojis function would look something like this :

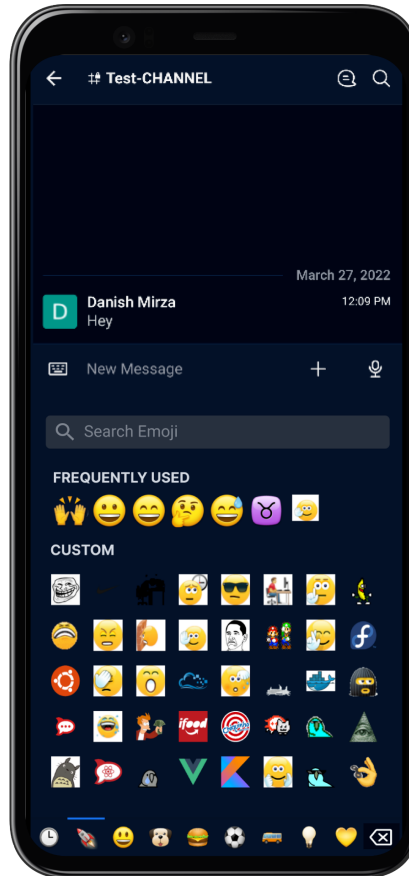
```
1  searchEmojis = debounce(async (keyword: any) => {
2    const db = database.active;
3
4    // Get custom emojis from the database
5    const customEmojisCollection = db.get('custom_emojis');
6
7    // Matches letters from any alphabet and numbers
8    const likeString = sanitizeLikeString(keyword);
9    const whereClause = [];
10   if (likeString) {
11     whereClause.push(Q.where('name', Q.like(`${likeString}%`)));
12   }
13   const customEmojis = await customEmojisCollection.query(...whereClause).fetch();
14
15   // Filter emojis that contains the keyword
16   const filteredEmojis = emojis.filter(emoji => emoji.indexOf(keyword) !== -1);
17   const mergedEmojis = [...customEmojis, ...filteredEmojis];
18   this.setState({ searchedEmojis: mergedEmojis || [] });
19 }, 300);
```

UI Mockups

Reactions Picker



Emoji Keyboard



Reactions List

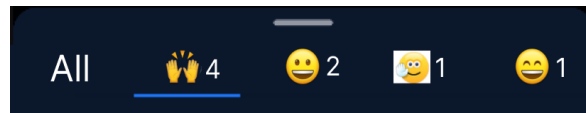
Another component that we want to migrate to the Action Sheet is the Reactions List. The current component lacks some features like displaying all the users who reacted (Only 3 usernames are visible at max for some reacted emoji) and opening his/her profile directly from this list is not possible.

The new proposed design improves the user experience by listing profiles of all the users at one place who reacted with some emoji. This makes it easier to check the users who reacted along with their name

and profile picture. Clicking on which, we'll be redirected to the user's profile.

The component I plan to use for the Reactions List is `<TabView/>` implemented using [react-native-tab-view](#).

- Scrollable tab bar :



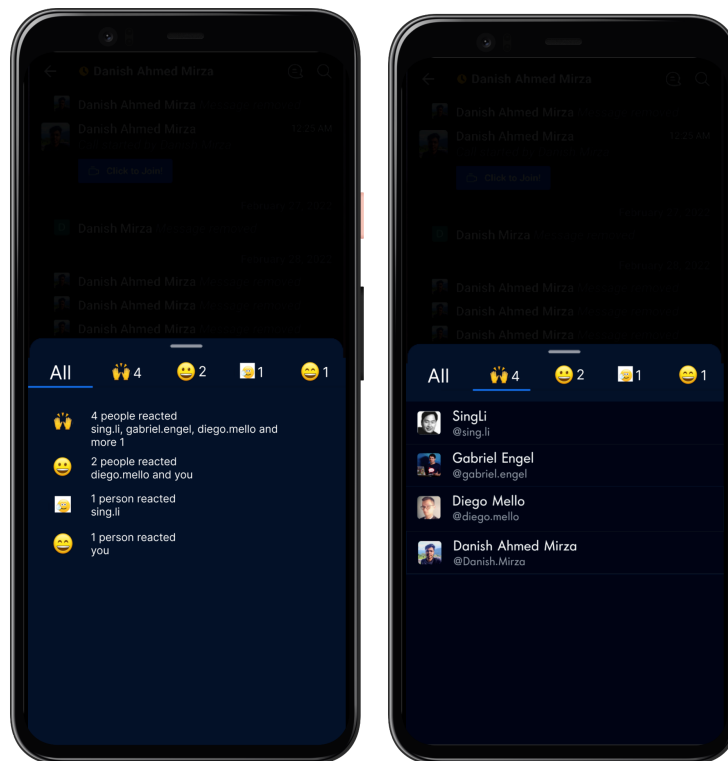
It'll list the emojis along with the number of users who have reacted with it. The 'All' section will list all the users who have reacted at one place.

- Users list :

It'll list the users who reacted with the current active emoji on the tab bar. Every user will be displayed with their profile picture, name and username.

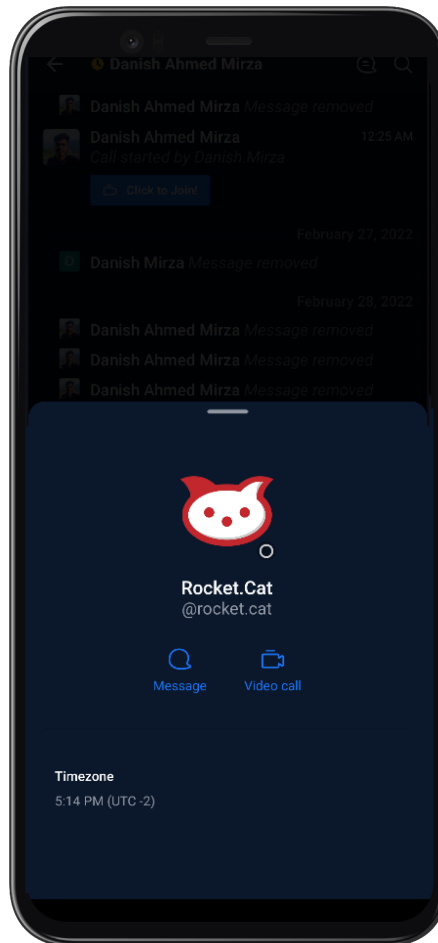
UI Mockups :

Reactions List



User Info View:

Clicking on a user from the reactions list will open this Action sheet.



Contributions to Rocket.Chat

- All the Issues opened by me can be found [here](#).
- All the PRs (Status: open) by me can be found [here](#).

Merged/Under review PRs in [Rocket.Chat.ReactNative](#) repository:

S. No.	Pull Request	Status
01	[FIX] Joining and leaving messages in teams	Merged
02	[FIX] Mention from suggestions concatenates to query string on autocomplete	Merged
03	[FIX] Unnecessary login dispatch on adding new server	Merged
04	[FIX] Disable tap gesture on call messages	Merged
05	[NEW] Delete my account feature	Under Review
06	[NEW] Update room item animations	Under Review
07	[NEW] Change font to Inter on both iOS and Android	Under Review
08	[FIX] Room actions not working in search view	Under Review
09	[NEW] Add an option to change message time format	Under Review
10	[IMPROVE] Add a clear button on search bar of new message page	Under Review
11	[IMPROVE] Display status of users in the members list of channel/team	Under Review

Timeline

Pre GSoC	<ul style="list-style-type: none"> • Work on the existing PRs and issues. • Discuss the deliverables and approach with the mentors. • Work on new features/bug fixes assigned to me.
Community bonding period (May 20 - June 12)	<ul style="list-style-type: none"> • Discuss and Finalize the design of the new Emoji Picker, Reactions List and other Action sheets. • Install and set up the new library. • Discuss the project with mentors and community to see if there can be more improvements in the design and user experience. • Read the documentations of <code>react-native-bottom-sheet</code> and <code>react-navigation</code>.
<u>Coding Period</u>	
Week 1-4 (June 13 - July 13)	<ul style="list-style-type: none"> • Write the code for the new Action sheet making it dynamic enough to render other custom components in it. • Migrate the existing Action Sheets to the new Action Sheet.
Week 5-6 (July 13 -July 25)	<ul style="list-style-type: none"> • Start the implementation of the new Emoji Picker component. • Work on the Emoji Picker component and complete all its functionality like searching and deleting emojis.
Phase 1 Evaluation	

Week 7-10 (July 25 - August 25)	<ul style="list-style-type: none">• Rendering the Emoji Picker component in the Action Sheet for the Reaction Picker. Also updating the component in the Emoji keyboard to this new Emoji Picker.• Implement the new design of Reactions List.
Week 11-12 (August 25 - September 5)	<ul style="list-style-type: none">• Complete the Reactions List functionalities like navigating to the user's profile from the users list and rendering the component in the Action Sheet.
Final week (September 5 - September 12)	<ul style="list-style-type: none">• Submit final code and complete the project
Final Evaluation	

Commitments

I plan to devote around 40-45 hours (Or more) per week on the project and can work full time on the weekdays. I'm usually available between **10 AM IST (4:30 AM UTC / 1:30 AM BRT)** to **2 AM IST (8:30 PM UTC / 5:30 PM BRT)** for any type of communication/meetings/reports. I have no other commitments during the GSoC and I'm applying **only to Rocket.Chat** for this Google summer of code 2022.

Post GSoC

I've been a Rocket.Chat user for over 5 months now and just love its feature- rich interface , amazing community, great vision, and lots of scope to learn. I plan to continue contributing to it even after the GSoC period and would love to add value to the community in every way possible.

Thank you