

COP290: Design Practices

User Registration App

Akshit Tyagi (2014EE10710)
Rishabh Kumar (2014PH10817)
Karan Dwivedi (2014CS10227)

15 January 2016

‘User Registration App’ is a simple Android app to test communication with a server using Volley JSON library. It takes in the names and entry numbers of team members through its user friendly GUI and sends it to the server when Submit button is pressed. We also check if the user types invalid (or blank) input.

The app also receives the server response and shows it to the user. In case the server sent a “Registration failed” message, the app prompts the user to try again later.

The code for the project is being maintained in this repository:
<https://github.com/akshittyagi/AssignmentZero>.

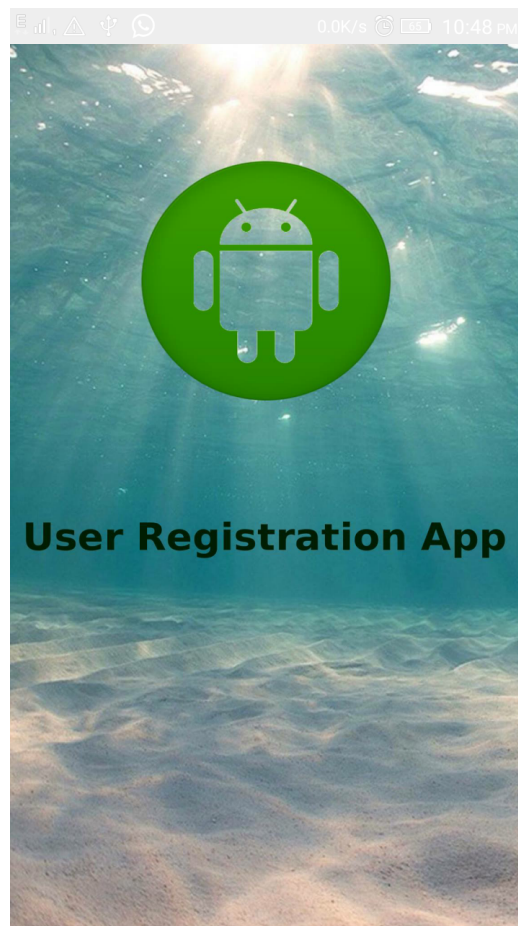


Figure 1: Splash Screen

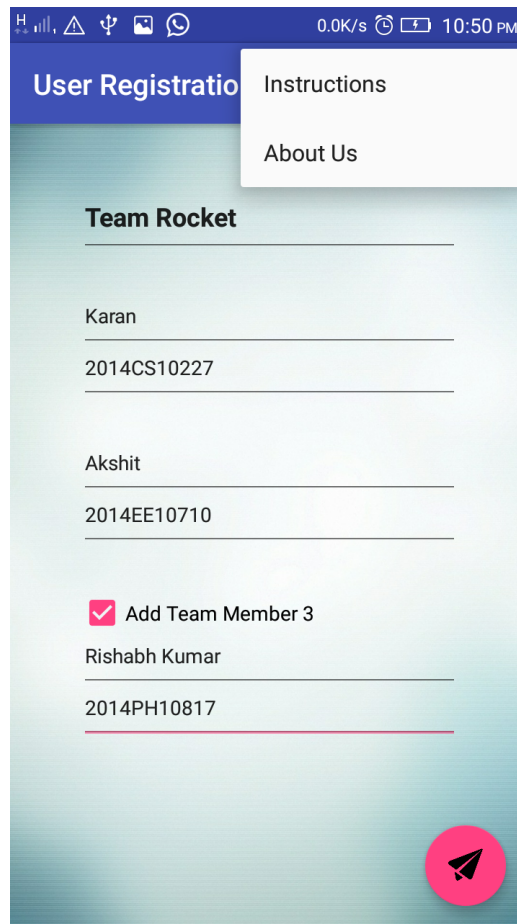


Figure 2: User Input Screen

1 User Interface

- **Details of the screens visible to the user**

When the user starts the app, (s)he is greeted by the splash screen. Then the user sees a screen with input fields, where he can input his Team name and team members' details. The activity has 6 input fields and a floating action button. To register on the web server, the user enters the input using inbuilt keyboard and taps the submit button. In case there is an error in the input or the server response, the user is prompted to change it.

- **Animations, buttons (enabled/ disabled under what conditions)**

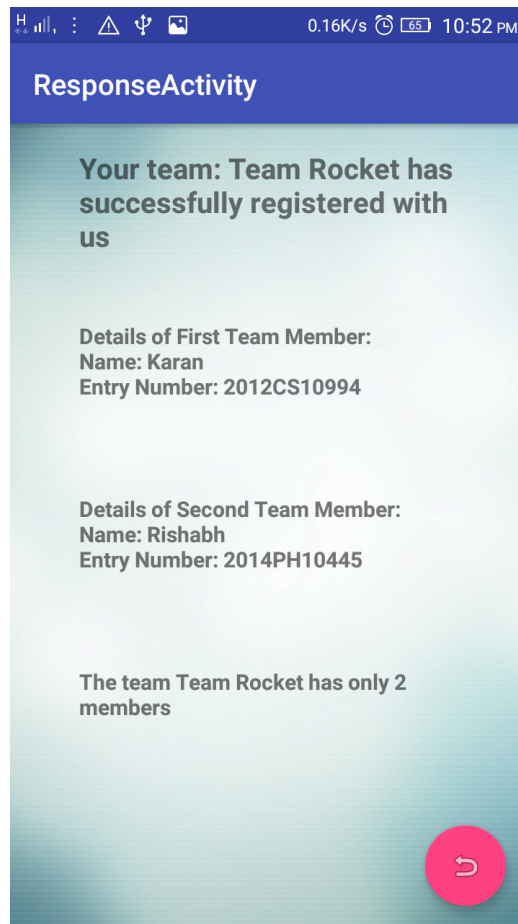


Figure 3: Server Response Screen

Following Google's material design principles, the screen has a floating submit button, which connects to the server only after the user has entered valid input in all fields. Otherwise, it highlights the incorrect inputs and prompts the user to correct them.

When the Check Box 'Team has 3 members' is checked, the third set of input fields becomes visible. On unchecking, it becomes invisible again.

- **Actions performed when a user enters information, presses a button/ icon etc.**

When the user taps any input field, the keyboard shows.

On pressing the submit button, the app checks the input and if found valid, sends it to the web server and receives its response.

This is shown to the user using a SnackBar on bottom of the page.

The user can also read the instructions and know about the team through the Action Bar.

2 Implementation Details

- **Organization of user information (Is it in a special User class, or is it distributed in arrays for each user entry)**

Data is stored in a Java Map object.

- High level functions/methods used and descriptions of each one

checkValidity: Checks the validity of text entered on pressing the submit button.

checkValidityName: Ensures name has only letters and is less than 16 characters long.

checkValidityEntryNumber: Ensures Entry number follows the format used in IIT Delhi and only valid years, department codes are entered

- Methods for network communication.

sendResult: Puts all data into a Map object and sends it using Volley RequestQueue object.

3 Error Handling

- Error Scenarios:

An error is shown when the user enters invalid Entry Number or enters non characters in Name. The length of Name is restricted to 16 to avoid issues with server response.

setError method is used to show a symbol and a help text to the user with relevant information.

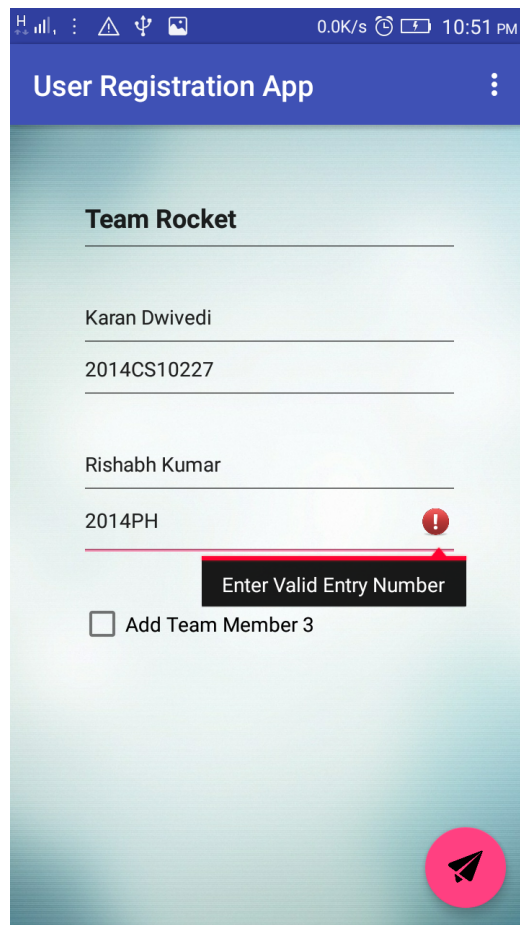


Figure 4: Error Handling