

[Description](#)

[Intended User](#)

[Features](#)

[User Interface Mocks](#)

[Screen 1 - Common Screen](#)

[Screen 2- Sign in Screen](#)

[Screen 3 - Speaker Dashboard Screens](#)

- [Screen 1- ProfileScreen](#)
- [Screen 2 - Follower Screen](#)
- [Screen 3- Notification Screen](#)
- [Screen 4 - Event Screen](#)

[Screen4 -Find Speakers.](#)

- [Screen 1- Follower Screen](#)
- [Screen 2- Search Screen](#)
- [Screen 3- Profile Screen](#)
- [Screen 4- Notification Screen](#)
- [Screen 5- Event Screen](#)

[Widget Screen](#)

[All Mockups](#)

[Key Considerations](#)

[How will your app handle data persistence?](#)

[Describe any corner cases in the UX.](#)

[Describe any libraries you'll be using and share your reasoning for including them.](#)

[Describe how you will implement Google Play Services.](#)

[Next Steps: Required Tasks](#)

[Task 1: Project Setup](#)

[Task 2: Firebase Setup](#)

[Task 3: Google Maps Setup](#)

[Task 4: UI](#)

[Task 5: Subtasks](#)

[Task 6 Data](#)

[Task 7: Notification](#)

[Task 8: Testing](#)

[Task 9: App Widget](#)

## App Name: **TechSpeakUp**

### Description

Write a brief summary of what your app does. What problem does your app solve?

TechSpeakUp App is designed to find all technical speakers in one place without the need to search anymore, we have them for you in 1 app, this app will help you to find speakers based on location, skill level, review, upcoming event based.

TechSpeakUp app solves one of the major issue which is "Finding all technical speaker in one place".

TechSpeakUp also includes couple of "Add on features" as listed below.

- Map displaying speakers based on location, skills, review.
- View Events details of the speaker.
- Helps to connect with other speakers, network and follow up any upcoming event.

These Add on makes the search more effective for people who are looking for speakers.

### Intended User

- Event organizer
- Students, Professors
- Tech Events, IT professions, Mentors, Google Developers, Youtubers.

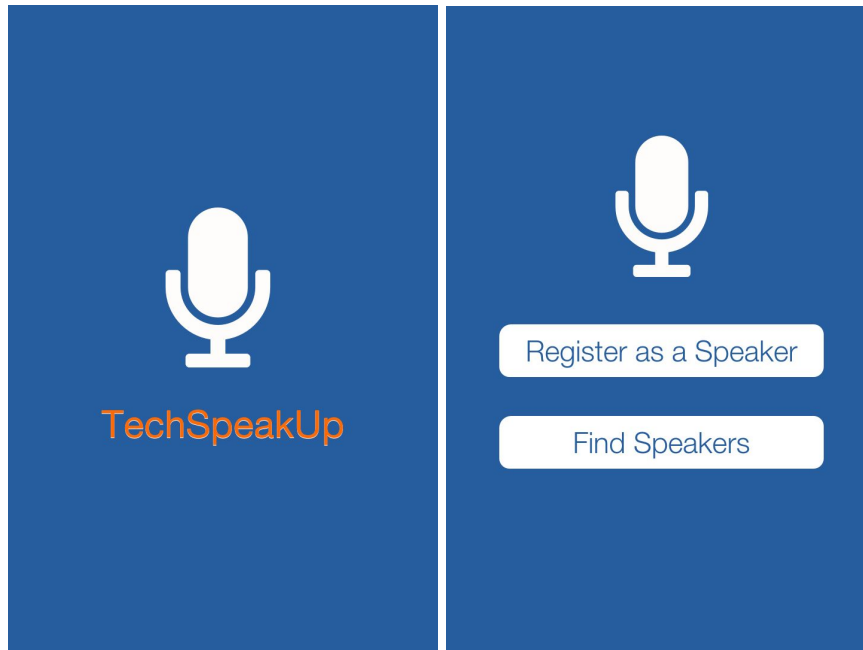
### Features

List the main features of your app. For example:

- Locate speakers.
- Speakers Register
- Connect with Speakers.
- Book them for events.

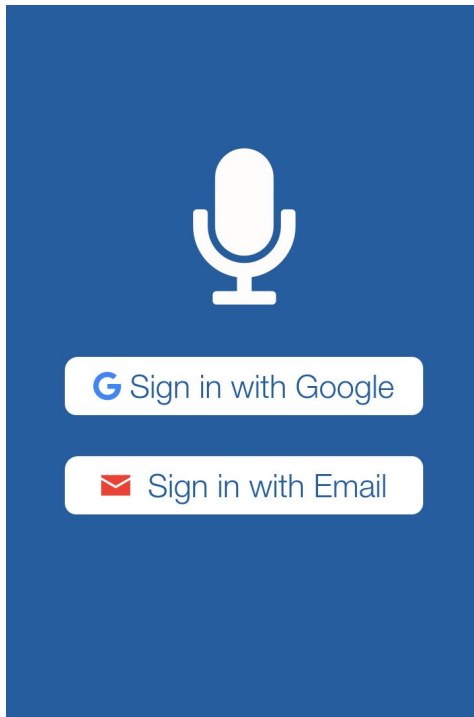
## User Interface Mocks

### Screen 1



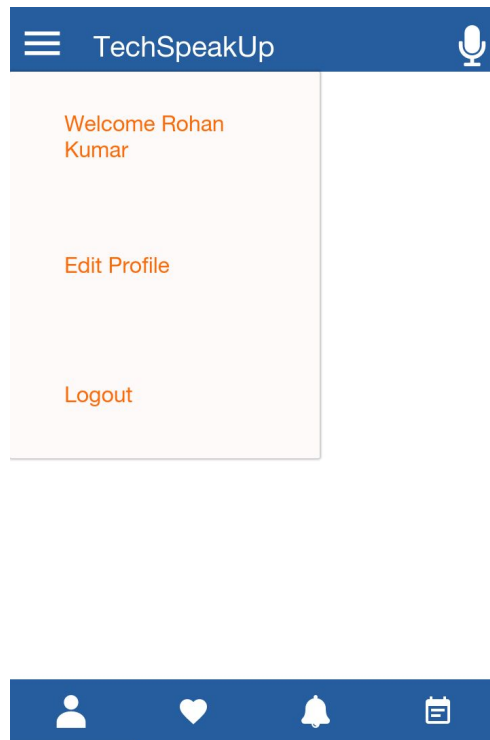
- This the First screen that gets displayed when you install for the first time.
- We have two Dashboards one for the speakers and one for those looking for speakers.

## Screen 2



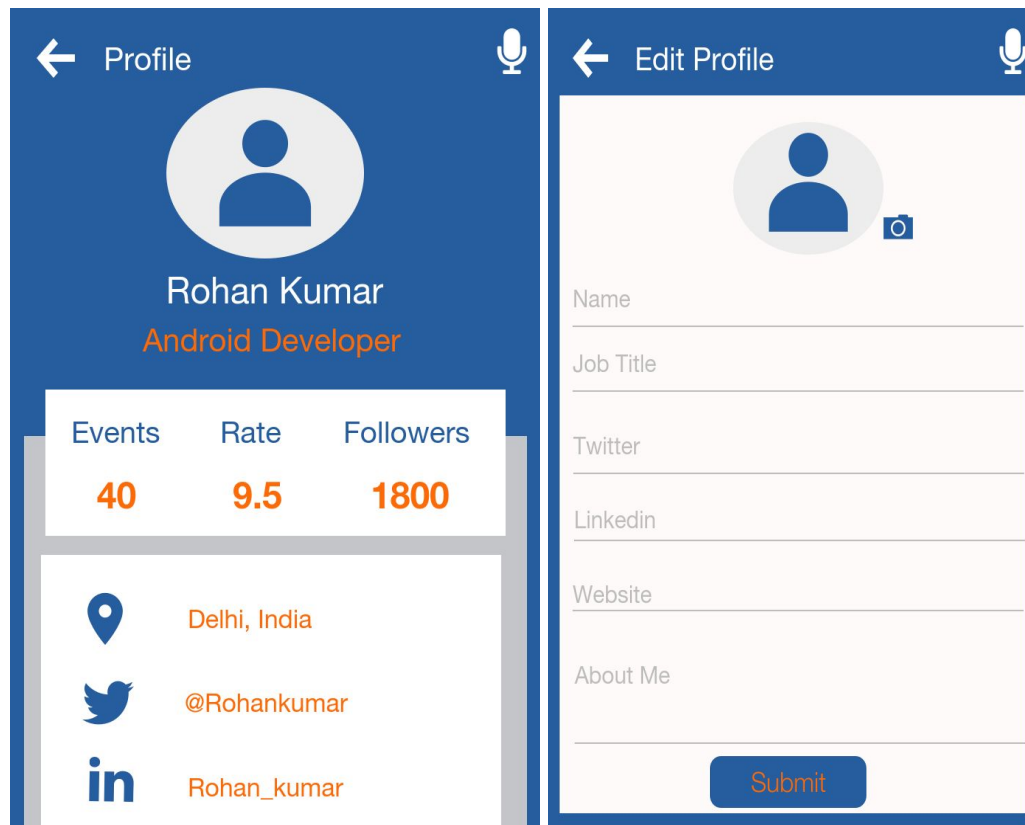
2. Second screen that prompts to register either using Google account or normal Email sign in and this would be done using Firebase Authentication.

## Speaker Dashboard Screens



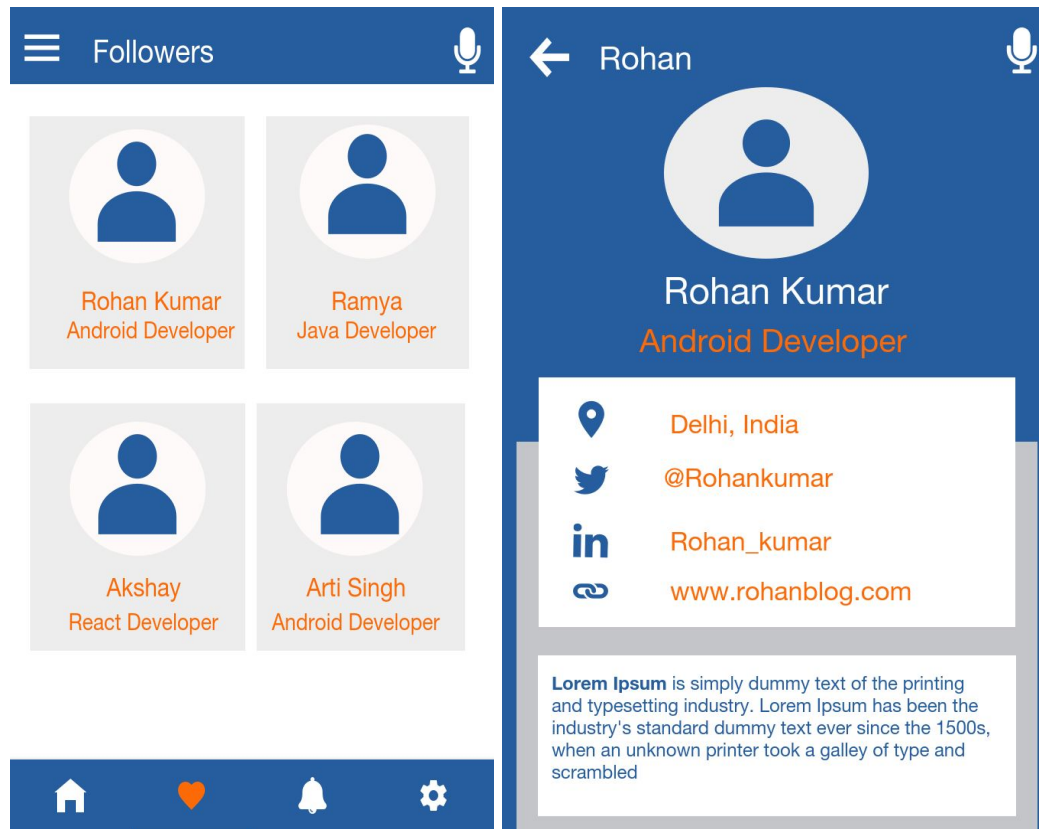
- We have two sections with menu one using hamburger icon on the left corner and another is the bottom navigation menu.
- Left to Right Menu (Profile, followers, notification and Events)
- Main Menu contains (Edit Profile and Logout Options)

## Profile Screens



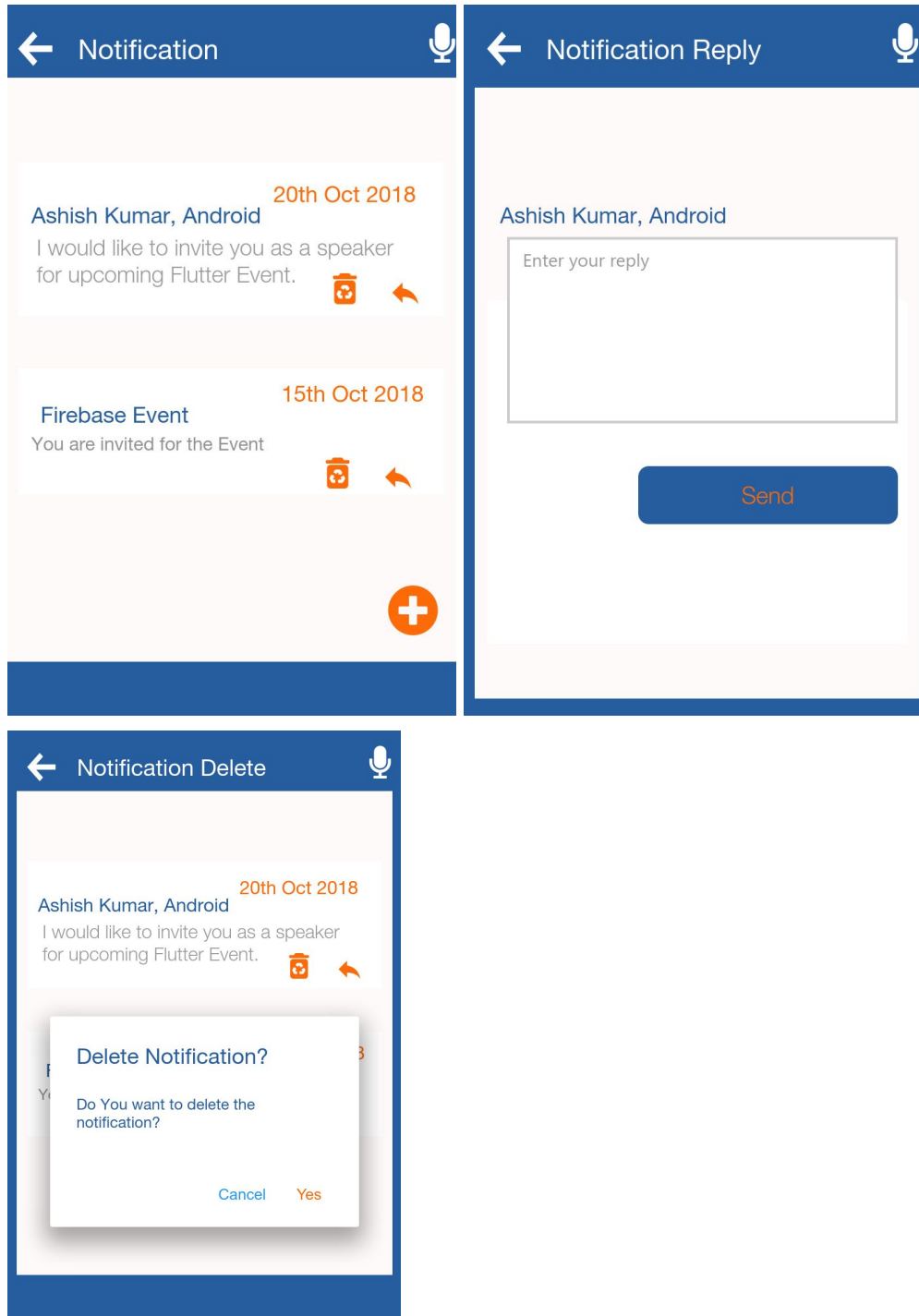
- This Section will help to view and edit the Speaker details, first screen display the Speaker details and second screen allows to update the Profile.

## Follower Screen



- Speaker can view the list of followers and when clicked on the Followers details the detail screen about the followers appear.
- First Screen - Follower list
- Second Screen - Follower details.

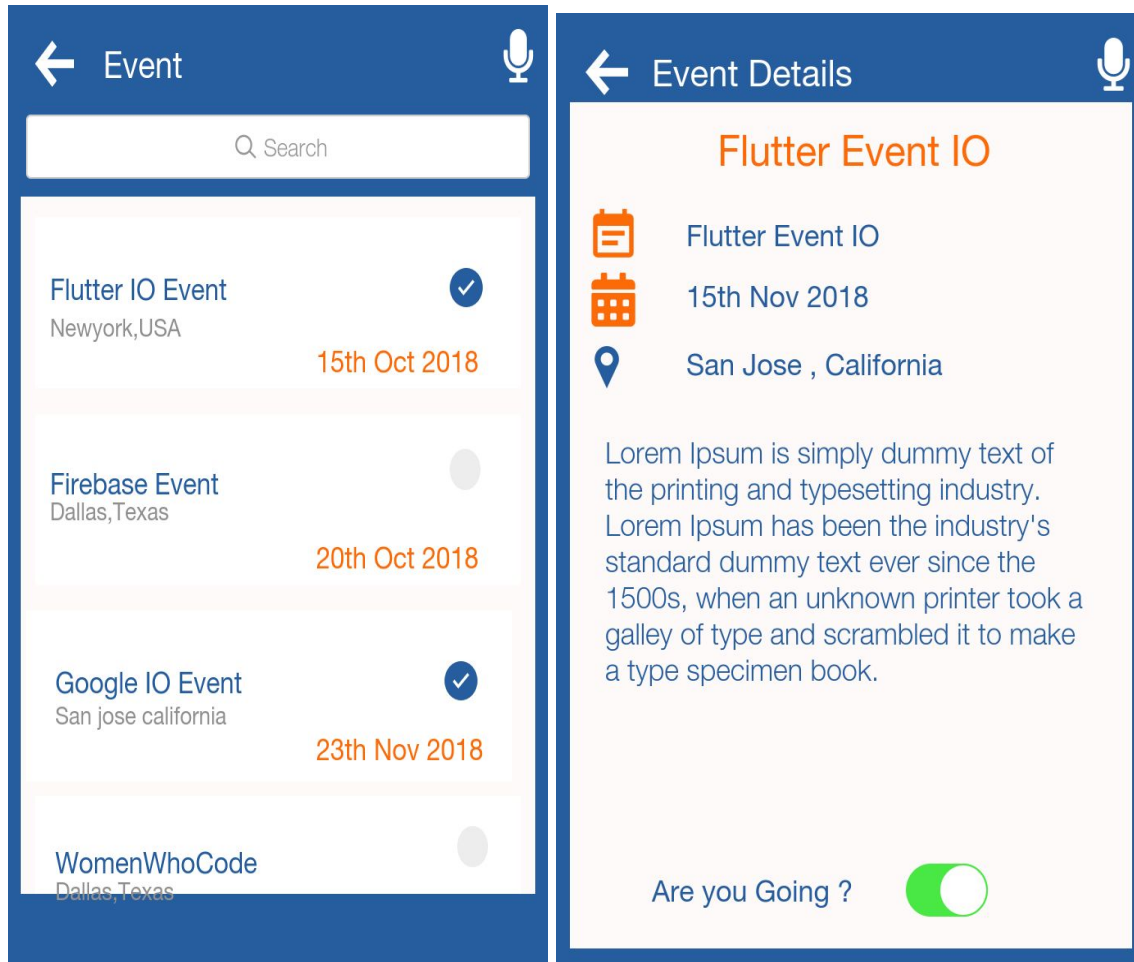
## Notification Screen



- First Screen gives details of the notification the speaker receives. Every Notification as a "Delete" and "Reply" icons.
- Second Screen appears when you click on "Reply" and you can respond.
- Third screen appears when click on the "Delete" icon which deletes the notifications.

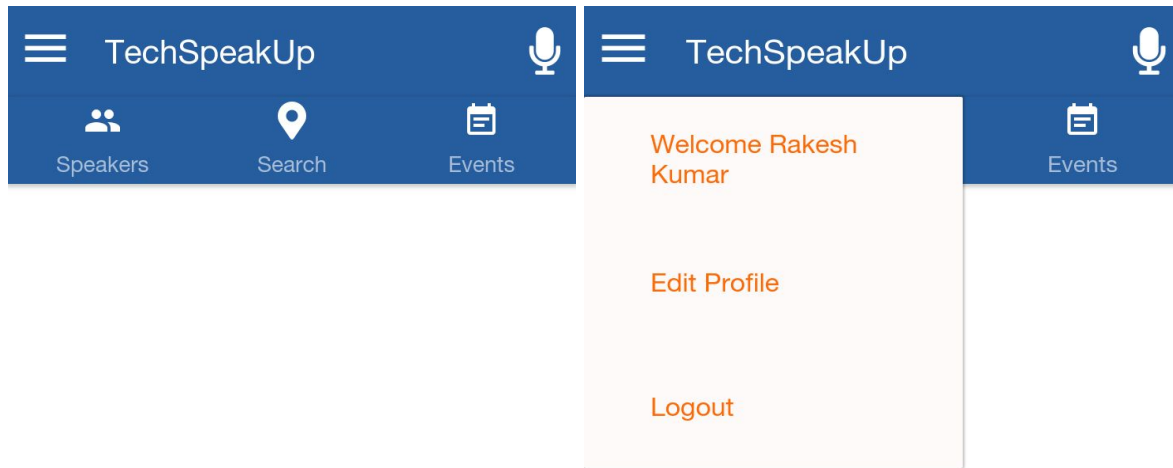


## Event Screen

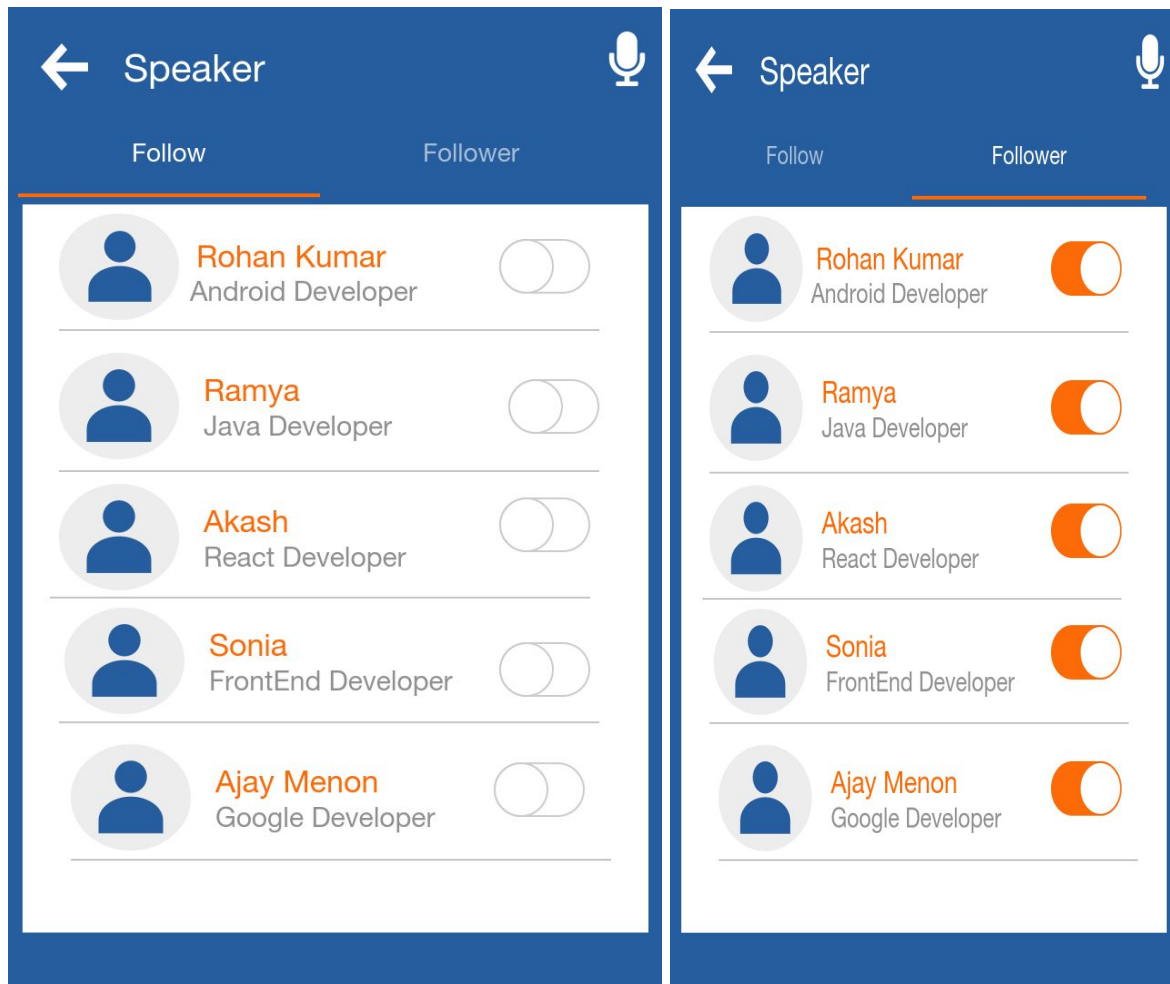


- First Screen gives the details of all the events occurring and you can also filter your search.
- Second screen gives the details of the individual event when clicked on it and also you have the provision to set if you going or not. This indication is only for your reference, this no where communicates to the event organizer.

## Find Speakers.

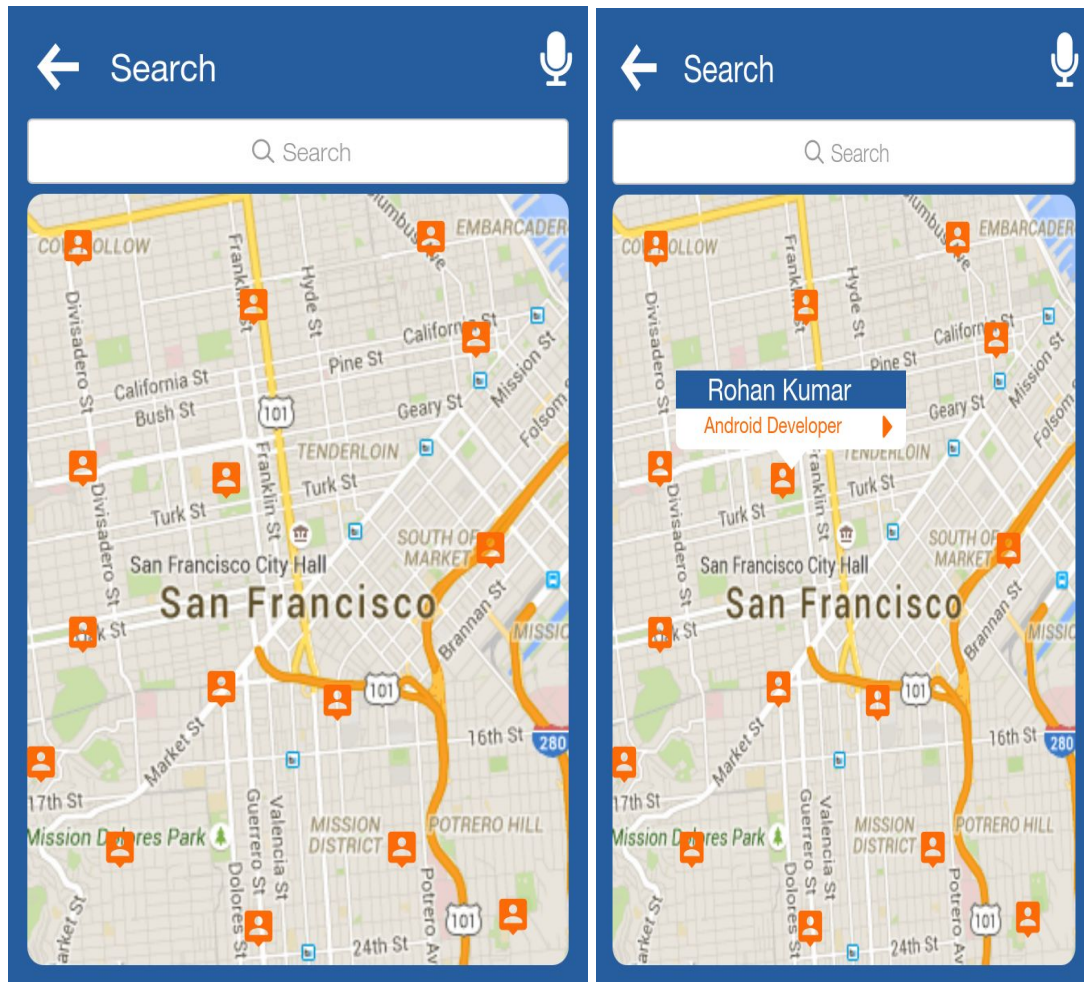


- This is the user Dashboard.
- We have 3 tab slider(speakers,search speakers,events)
- Bottom Navigation Menu bar (Home,profile,notification and settings)
- Menu items(Edit profile, Logout)

**Follower Details.**

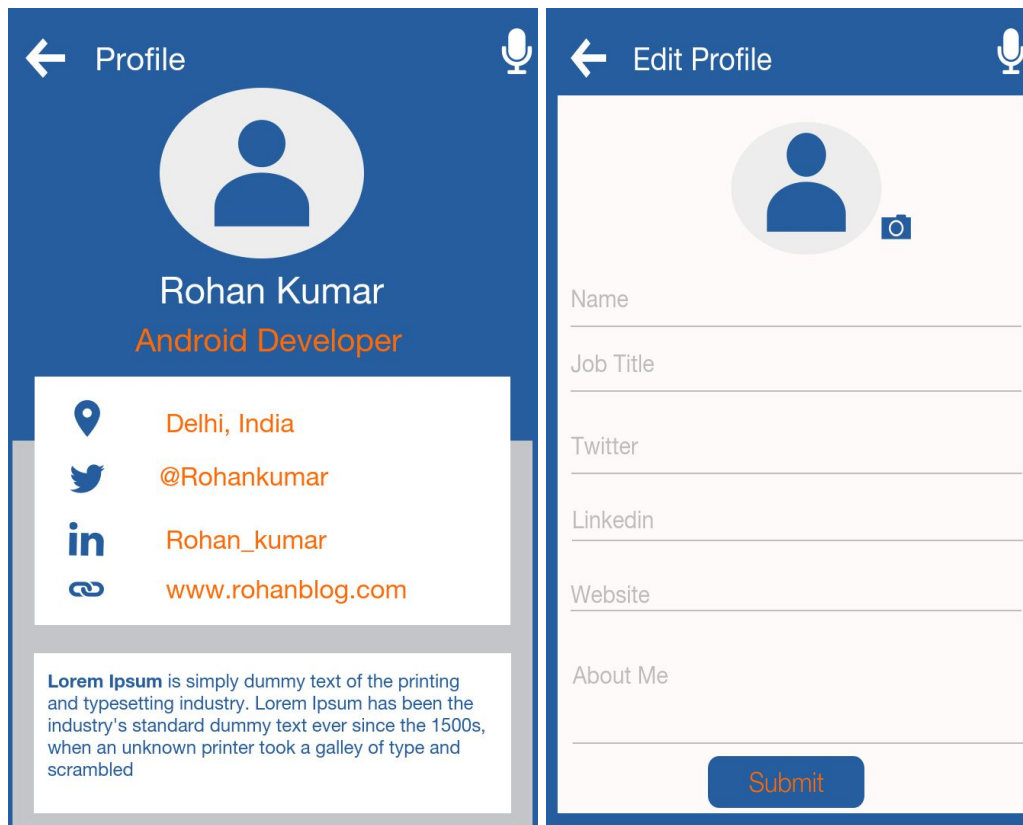
- This screen contains Tabbed layout with first one being “Follow screen” which display the list of Speakers which you can follow by click on the switch button.
- The Follower tab gives the list of followers you are following.

## Search Screen

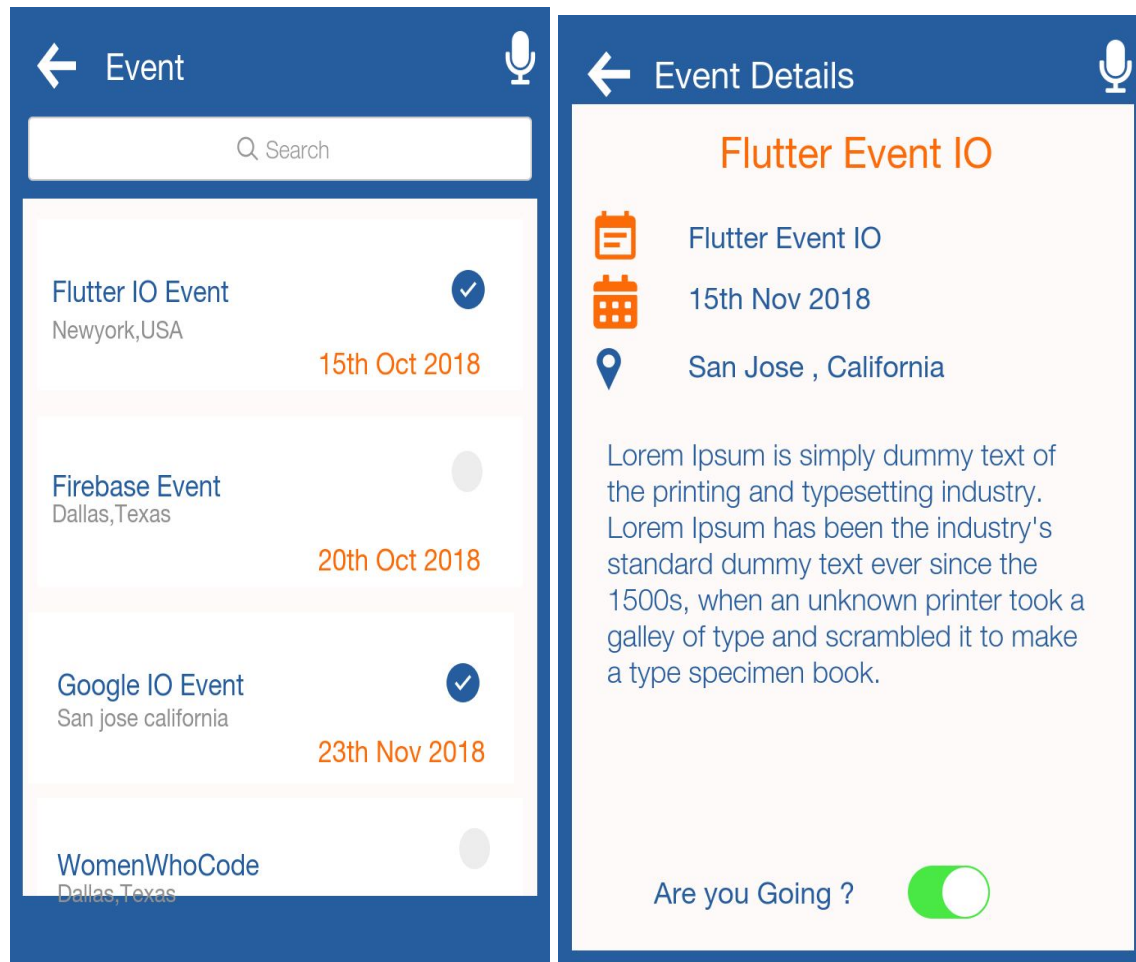


- This is Search Speaker screen that will allow you to search speaker based on your location and if you click on the map tooltip you get a brief detail about the speaker as shown in screen 2.
- On the second screen you can find the arrow when clicked it will navigate to the details screen of the speaker.

## Profile Screens.

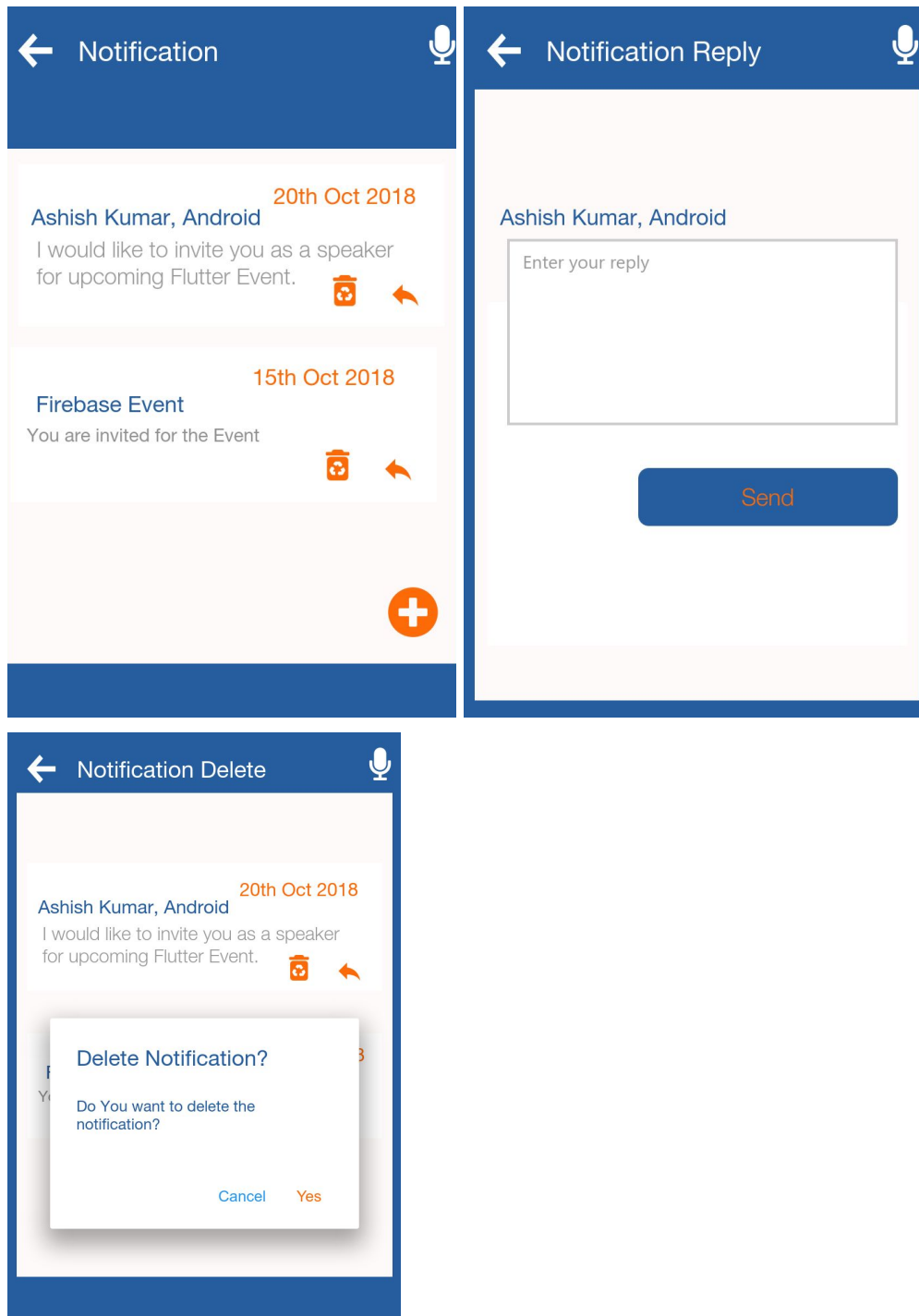


- This Section will help to view the profile details, first screen display your profile details and on the second screen you can update your profile.

**Event Screen.**

- First Screen gives the details of all the events occurring and you can also filter your search.
- Second screen gives the details of the individual event when clicked on it and also you have the provision to set if you going or not. This indication is only for your reference, this no where communicates to the event organizer.

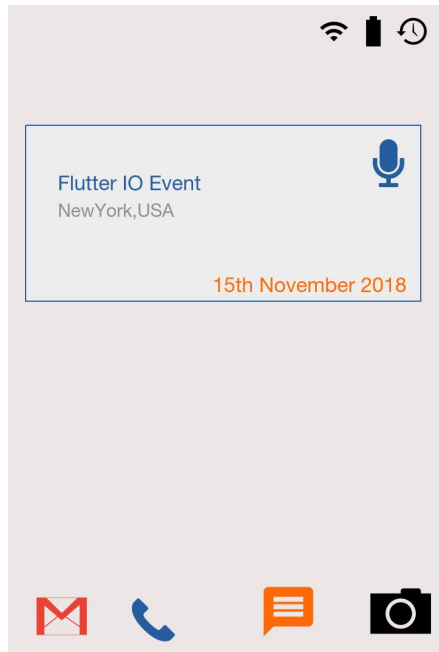
## Notification Screen.



- First Screen gives details of the notification the visitor receives. Every Notification as a "Delete" and "Reply" icons.
- Second Screen appears when you click on "Reply" and you can respond.

- Third screen appears when click on the “Delete” icon which deletes the notifications.

## WidgetScreen

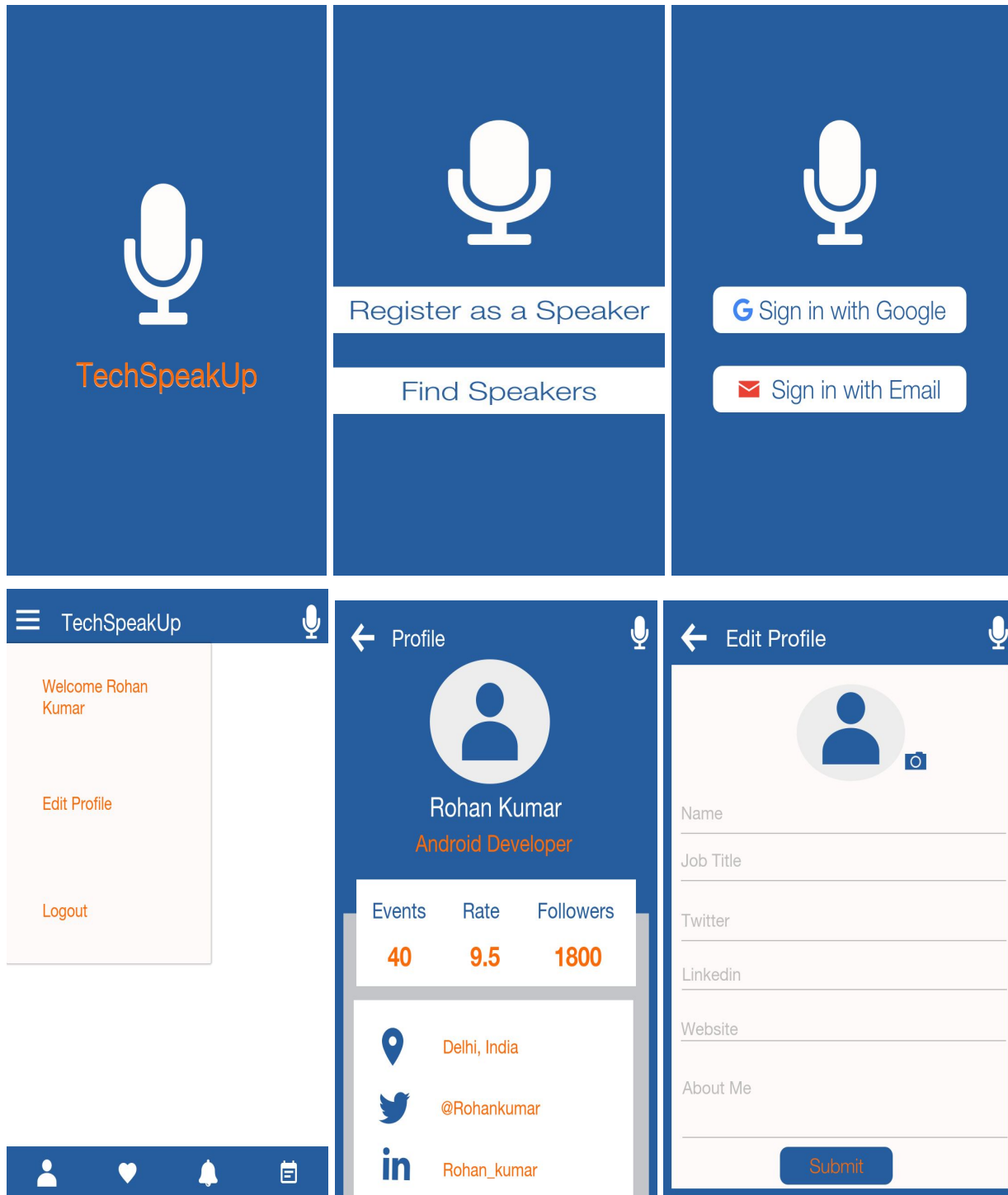


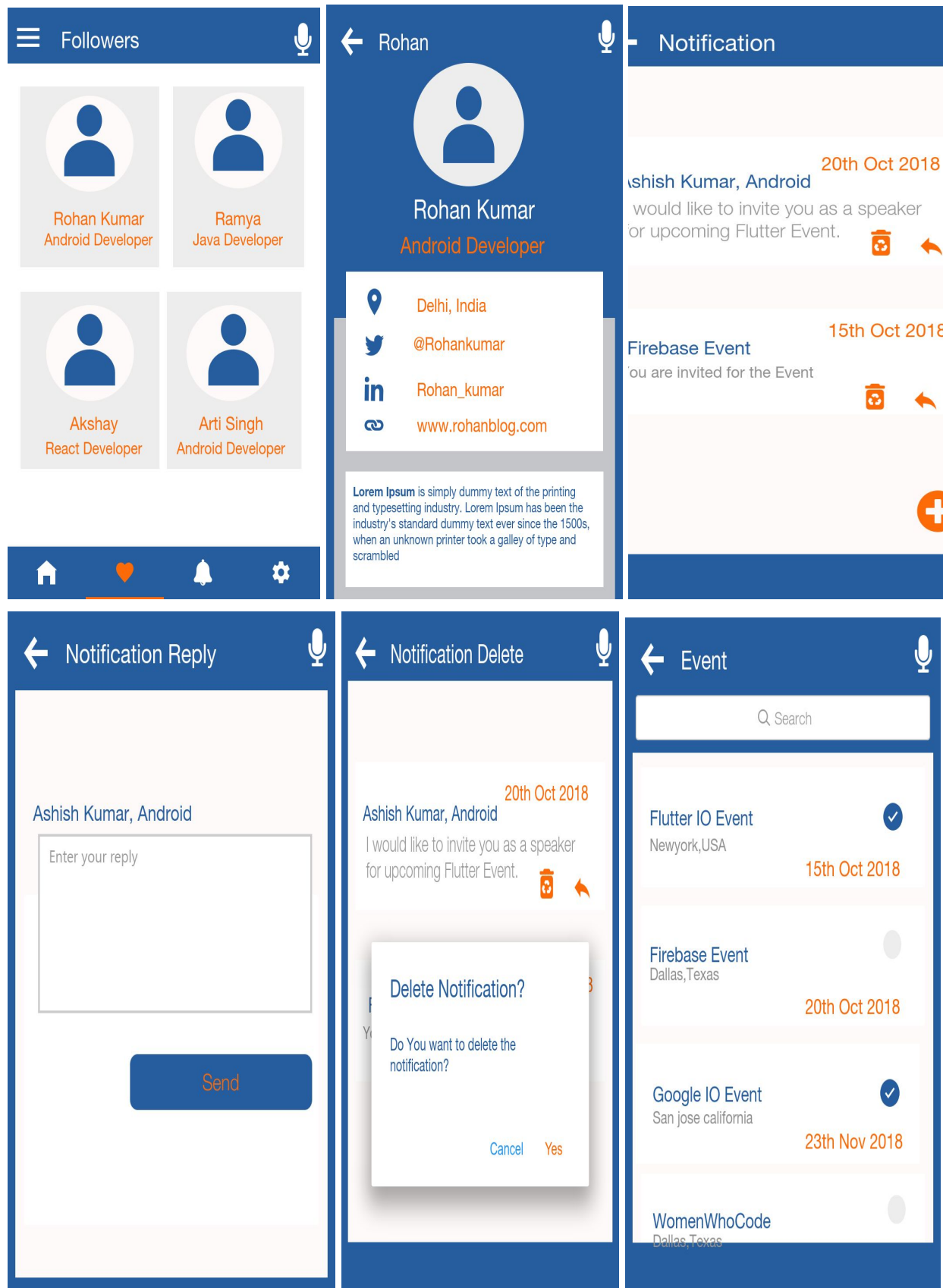
- The above screen application widget screen displays the latest upcoming event and when clicked it would be routed to the Event Section.

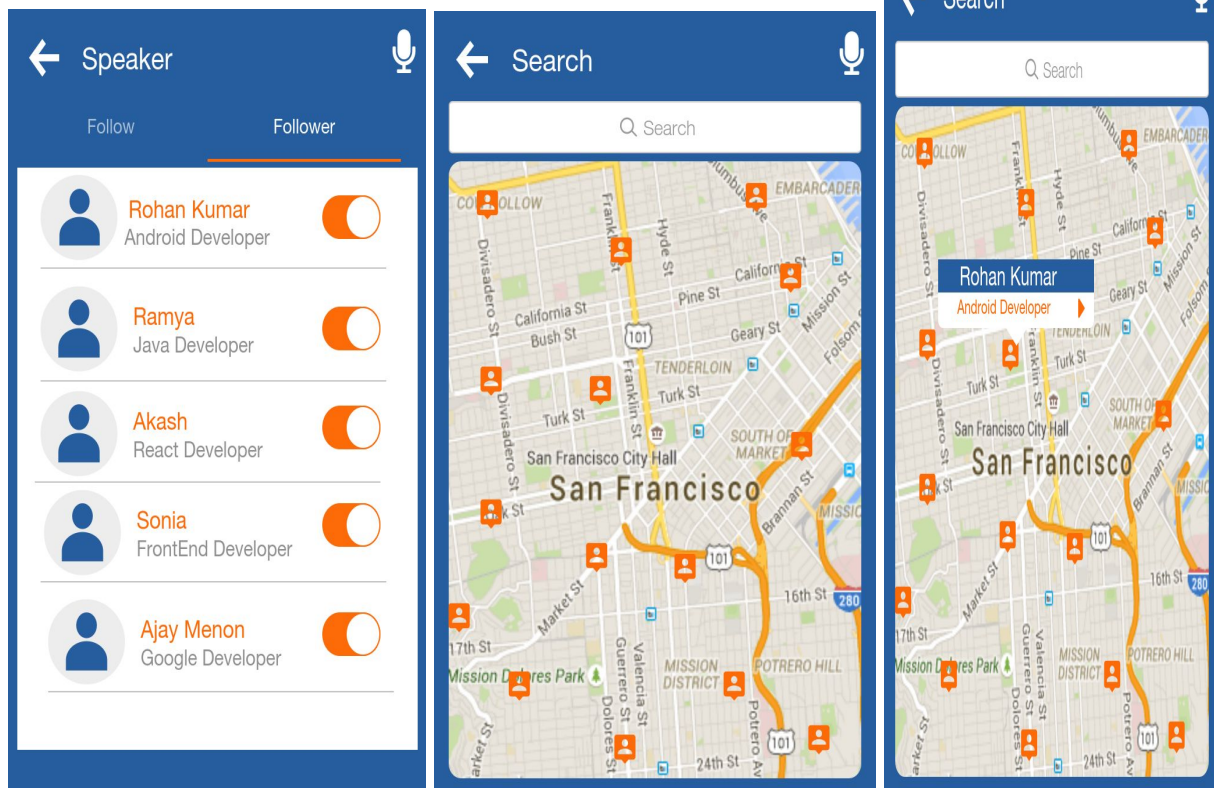
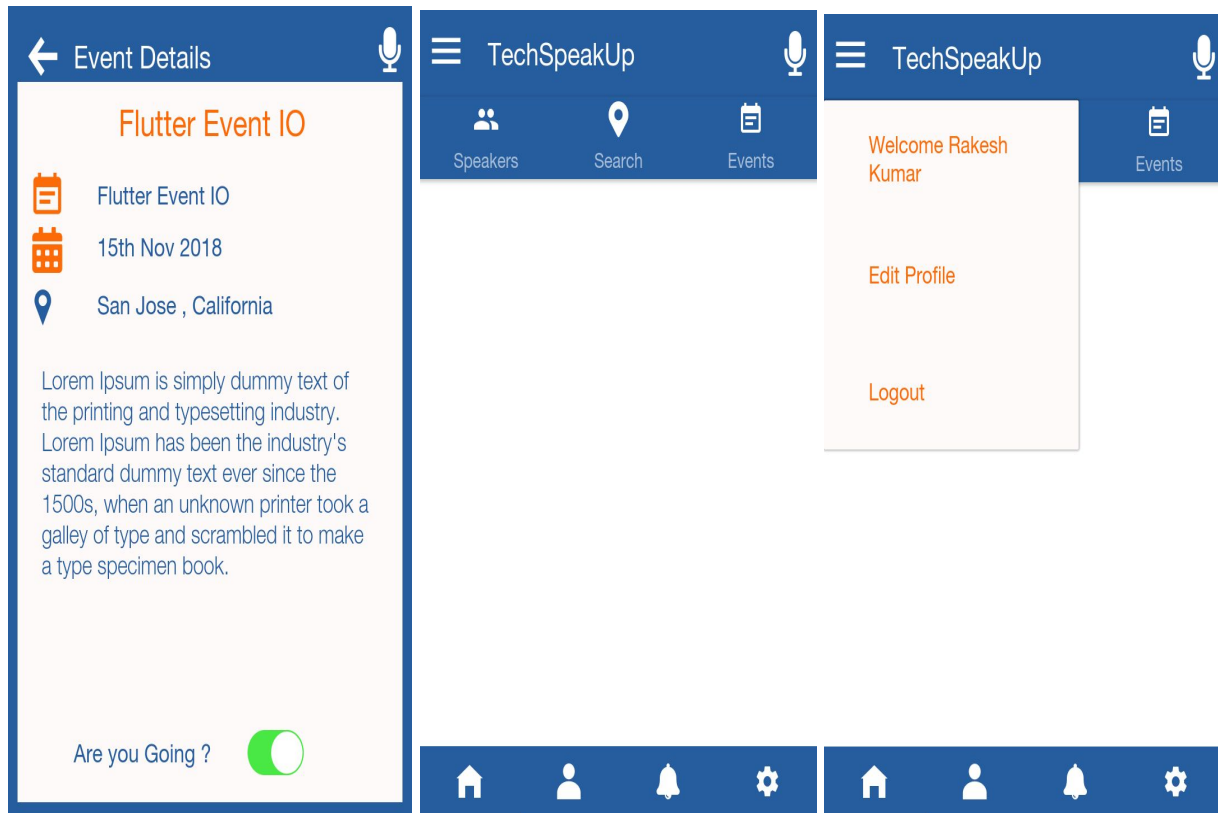





## All MockUI










← Profile




Rohan Kumar  
Android Developer

 Delhi, India


 @Rohankumar

 Rohan\_kumar

 www.rohanblog.com

Lorem Ipsum is simply dummy text of the printing and typesetting industry. Lorem Ipsum has been the industry's standard dummy text ever since the 1500s, when an unknown printer took a galley of type and scrambled

← Settings



Name

Job Title

Twitter

Linkedin

Website

About Me

Submit

← Event

Flutter IO Event

Newyork,USA

15th Oct 2018

Firestore Event

Dallas,Texas

20th Oct 2018

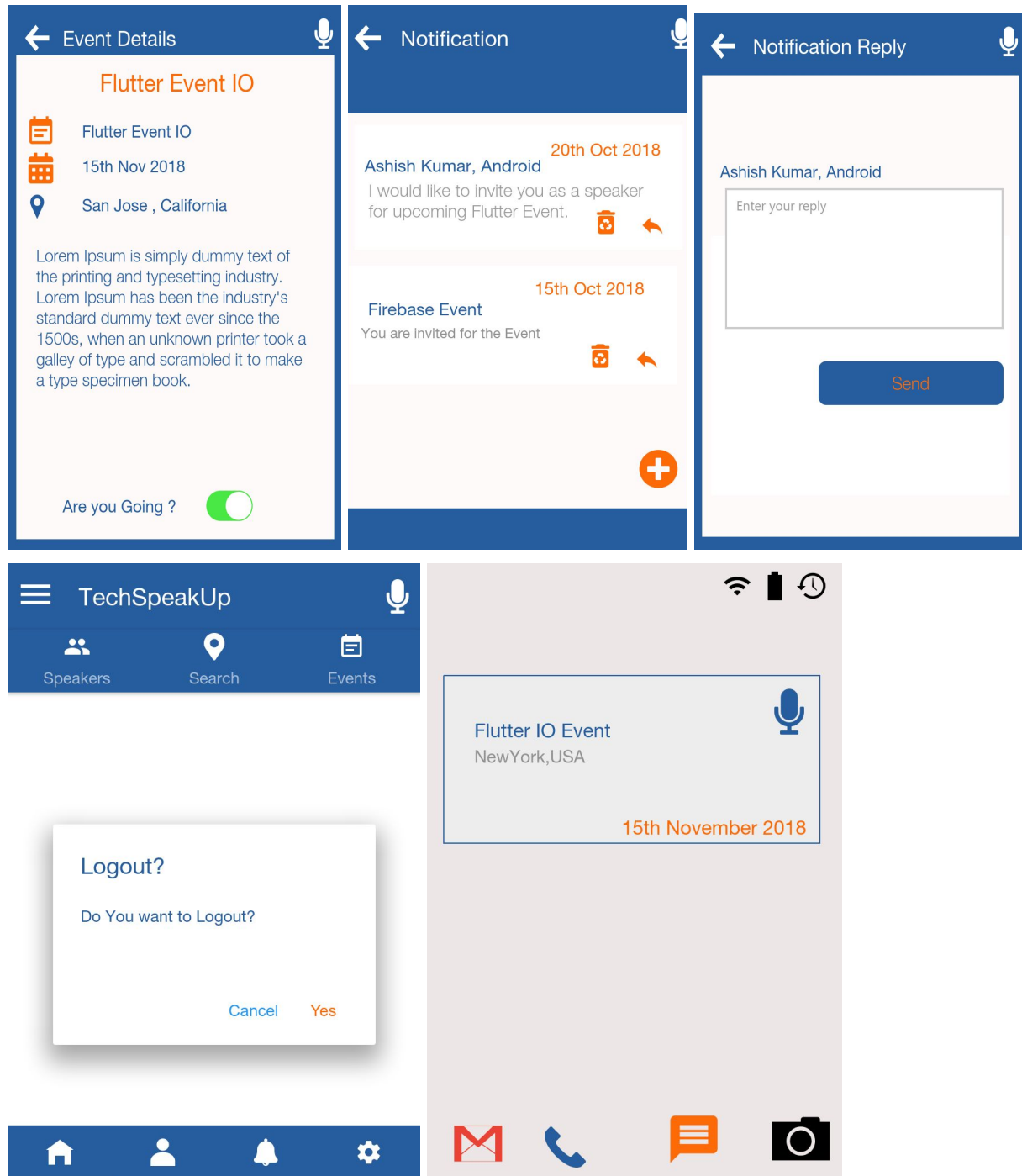
Google IO Event

San jose california

23th Nov 2018

WomenWhoCode

Dallas,Texas



## Key Considerations

- This app would be solely designed Using Java language.
- App will use the stable release versions of all the libraries,Gradle and Android Studio.
- Firebase JobDispatcher for Scheduling background jobs.
- App keeps all strings in a strings.xml file and enables RTL layout switching on all layouts.
- This app will ensure it includes support for accessibility by following the below rules.
  1. Make sure to provide “Alternative text” and “Content Description”.
  2. Provide descriptive label for the interactive elements.

### How will your app handle data persistence?

I am planning to use Firebase Realtime Database to store all the speaker and Visitor data.

### Describe any edge or corner cases in the UX.

If User goes offline use of firebase Realtime database caching features to preserve some user data until connectivity is regained.

User clicks back - while editing profile, or responding to notification , if back is pressed stating asking if they would like to discard.

### Describe any libraries you’ll be using and share your reasoning for including them.

Android support libraries for compatibility and material design.

- com.android.support:appcompat-v7
- com.android.support:design
- com.android.support:support-compat
- com.android.support:support-v4

**Picasso** - for handling image loading. (Picasso---'com.squareup.picasso:picasso:2.71828',)

**Firebase** - Sign In and Realtime Database.(Core ---- 'com.google.firebase:firebase-core:16.0.4', Database ---'com.google.firebase:firebase-database:16.0.3',Auth --- 'com.google.firebase:firebase-auth:16.0.4')

**CircleImageView** - For Circle images.([de.hdodenhof:circleimageview:2.2.0](#))

**Ripple Effect** - For ripple effect when button clicked.([com.github.traex.rippleeffect:library:1.3](#))

**ButterKnife** - Boilerplate code('com.jakewharton:butterknife-runtime:9.0.0-rc1')

Describe how you will implement Google Play Services or other external services.

- I will use Firebase Authentication for login.
- I will use Firebase Realtime Database for data storage.
- I will be using Google Location/Google Maps for speaker location and also display clusters of speakers.
  - com.google.android.gms:play-services-maps – for getting map
  - com.google.android.gms:play-services-places – for place info and latitude and longitude
  - com.google.android.gms:play-services-location - for getting location

## Task 1: Project Setup

1. Create Android Project, Github setup
2. Collate all the required libraries.

## Task 2:Firebase Setup

1. Create project on firebase console.
2. Add required dependencies in the project.
3. Set the Realtime Database.
4. Set FCM for notification.
5. Set Rules and Authentication as required.

## Task 3: Google Maps Setup



1. Google Play service SDK with Google Repository .
2. Get the Google maps Api Key.

#### **Task 4: UI**

1. Make sure all the resources are set, like icons, colors, images, style.
2. Start creating UI mockups using Material design guidelines and make it responsive.
3. Usage of proper layouts for all the screen to ensure it is looks fine in both landscape and portrait mode.

#### **Task 5: Subtasks**

1. Common screen
2. Speakers Dashboard and corresponding screens
3. Visitor Dashboard and corresponding screens
4. Dialogs.menu,notification

#### **Task 6: Data**

1. Query Database for retrieving and updating speaker details.
2. Also getting speaker detail based on their location and role type.

#### **Task 7: Notification**

1. Implement notification feature using Firebase FCM. which will help to send notification when a new event occurs.

#### **Task 8: Testing**

1. Creating test case to make sure the app is full functional as expected without crashing.

#### **Task 9: App Widget**

- Application widget will display the latest upcoming events and when clicked on it will be routed to the Event section.

---

#### **Submission Instructions**

- After you've completed all the sections, download this document as a PDF [ File → Download as PDF ]
  - Make sure the PDF is named "**Capstone\_Stage1.pdf**"
- Submit the PDF as a zip or in a GitHub project repo using the project submission portal

If using GitHub:

- Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
- Add this document to your repo. Make sure it's named "**Capstone\_Stage1.pdf**"