GitHub Username: federicomartini

Get Together

Description

Have you ever got together with your friends and you found yourself in a terrible mess to organize an event such as collecting taking a list of pizza for each guest or taking care of people requests coming from different social networks, instant messaging apps or by phone calls? You can rest easy now, because with Get Together you'll have your organization under control because it takes care of any boring aspect of organizing an event. The only thing you'll have to do will be just think up what to do and where to go with your friends!

Intended User

This app is intended for everyone who wants to organize an event for 2 or more people.

Features

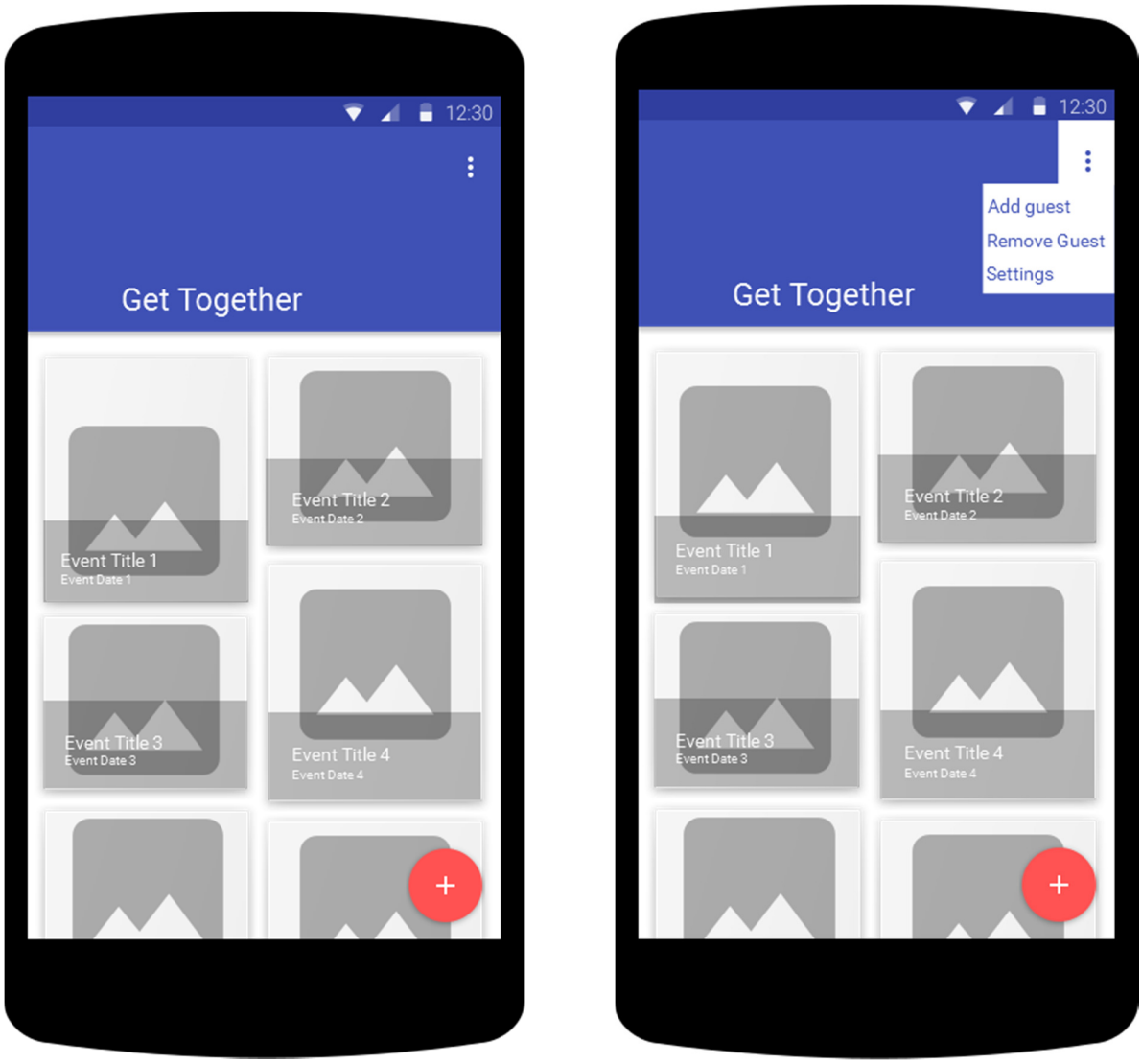
This app will:

- Save guests information
- Save events information
- Leverage of GPS sensor to enhance the Google Places experience
- Leverage of GPS sensor to provide information through Google Map
- Provide a Widget to directly acces to relevant information from the Home Screen
- Search events by keywords in the local database
- Share relevant information with your friends
- Import guests for an event by copying and pasting 2 or more messages from Whatsapp's chat by using the format by copying multiple messages.

User Interface Mocks

These can be created by hand (take a photo of your drawings and insert them in this flow), or using a program like Photoshop or Balsamiq.

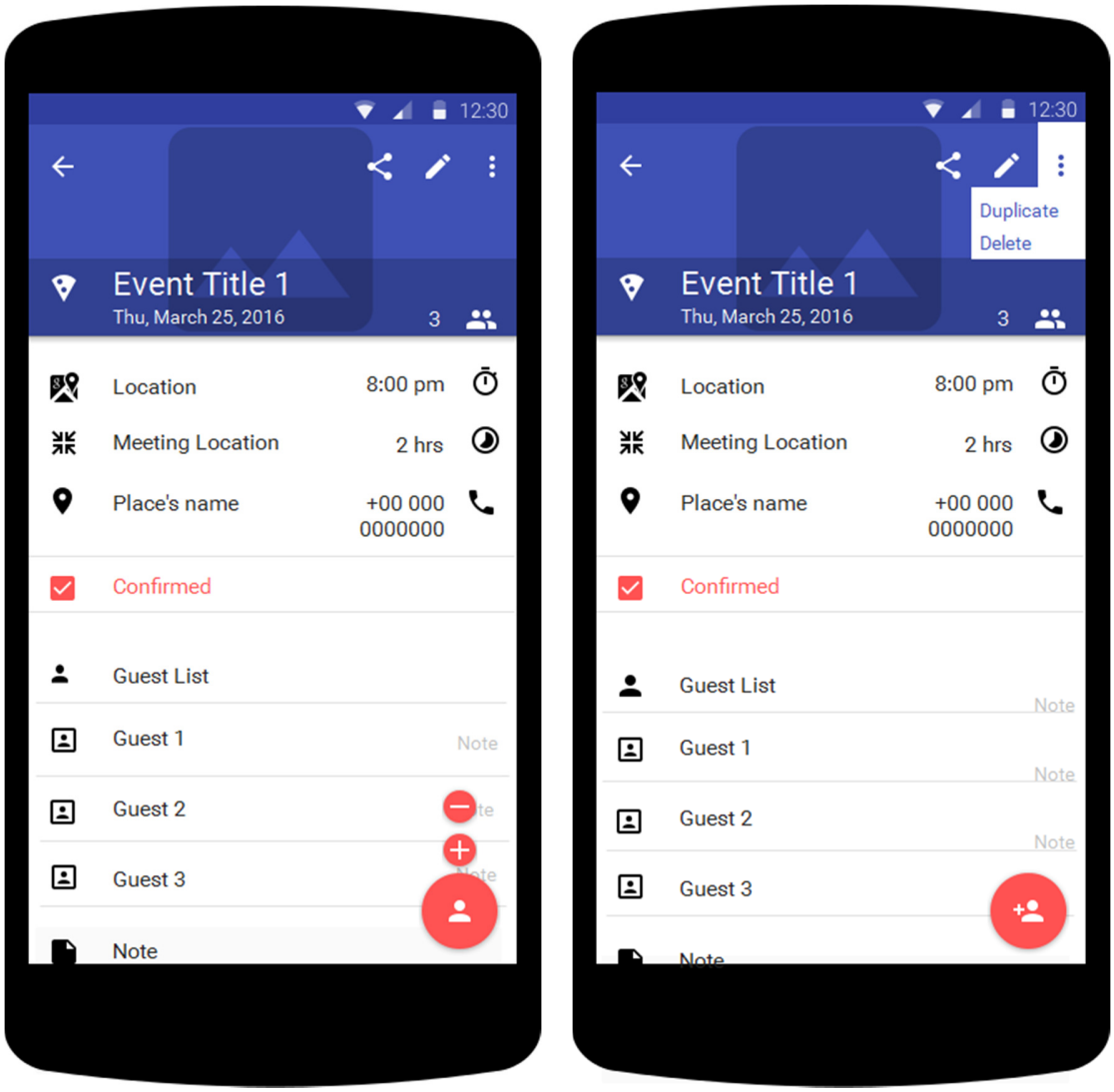
Screen 1



Main screen opened when the app is running on smartphones.
From this screen the user is able to:

- See each event created and arranged in a grid
- Tap on a event to reach its detail view
- Add an event by tapping on the FAB
- Open the overflow menu to Add guests or access to the App settings
- The page uses a collapsable App bar

Screen 2



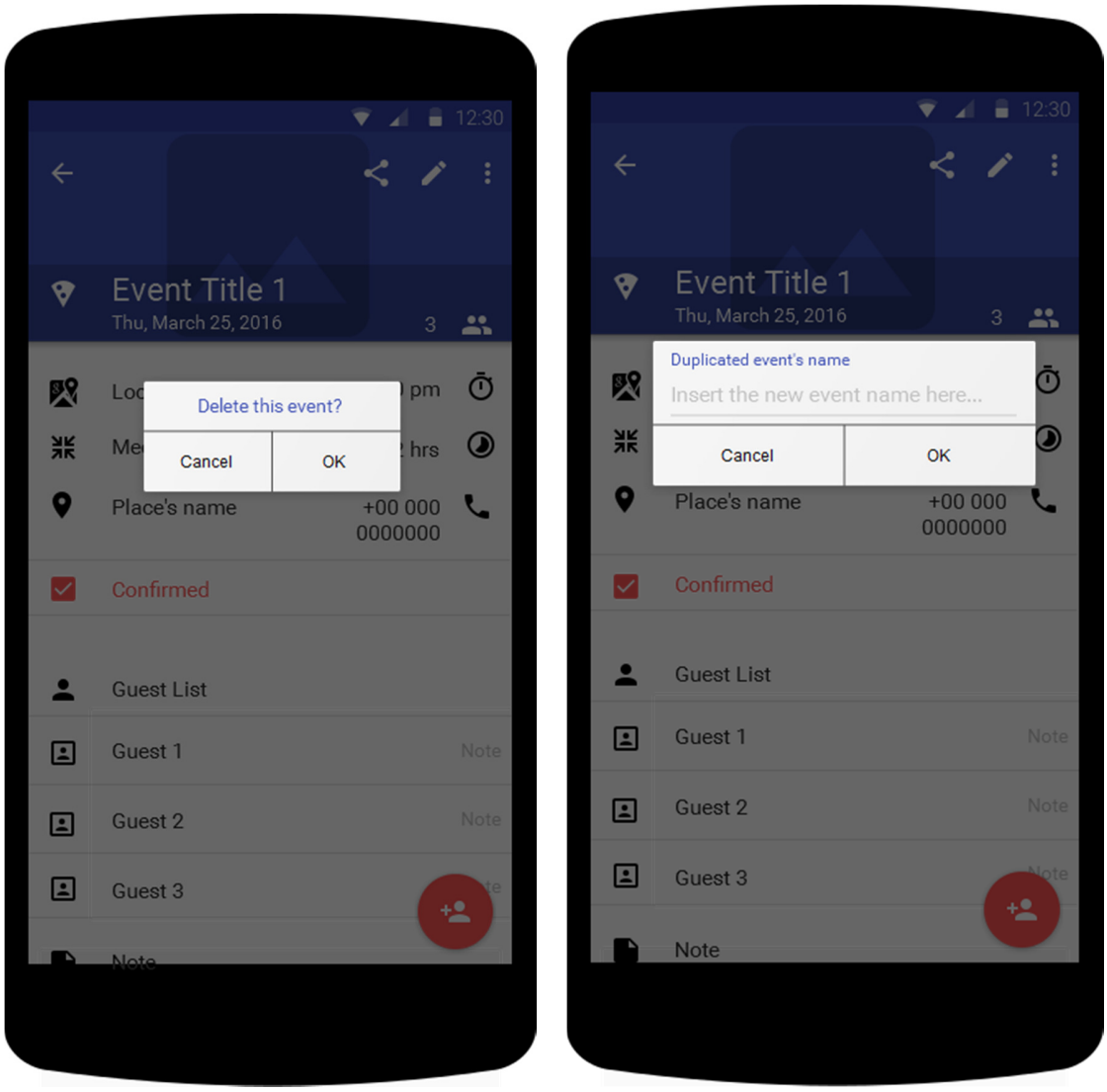
This is the Event Details screen and it mainly shows information about a tapped event from the Main Screen (Screen 1).

From this page, the user can check locations, time, duration, event's type, phone number of the Event Location, Guest List and some notes about the event. From here the user can Add or Remove guests or change some information's event. Furthermore, it's also possible to share a static message with people through Whatsapp to inform them about the status of the event.

By workflow menu's options the user can Delete the event or duplicate it by using same information except its Title that has to be new.

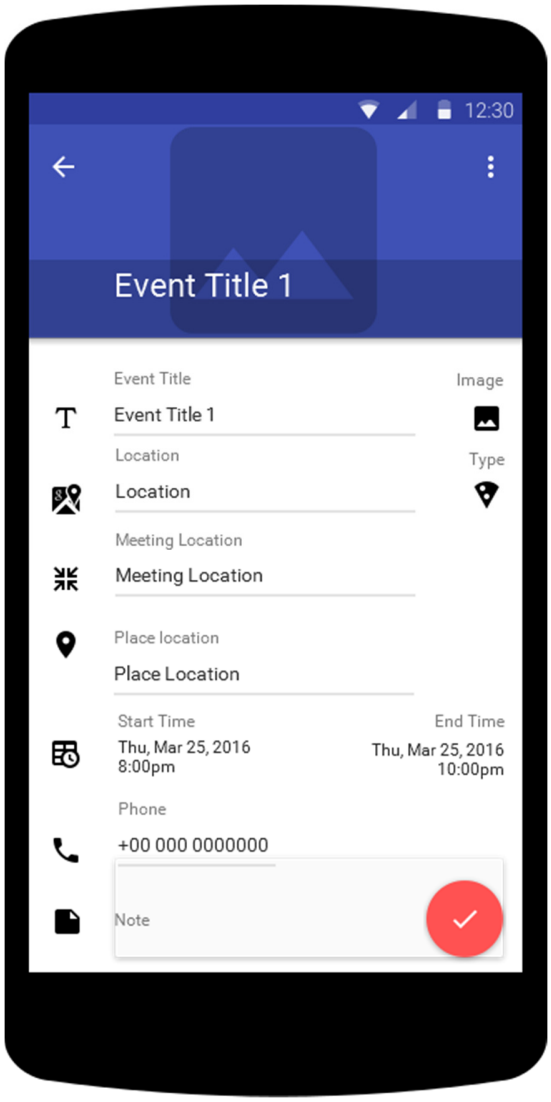
The Collapsable App Bar will be filled with a background image which cames from an image chosen by the User when the event was created/edited.

Screen 3



This are the kind of screens reachable when the user tap on the Event Details' overflow menu options. A dialog will be shown in order to allow the user to choose what to do next before effectively doing something.

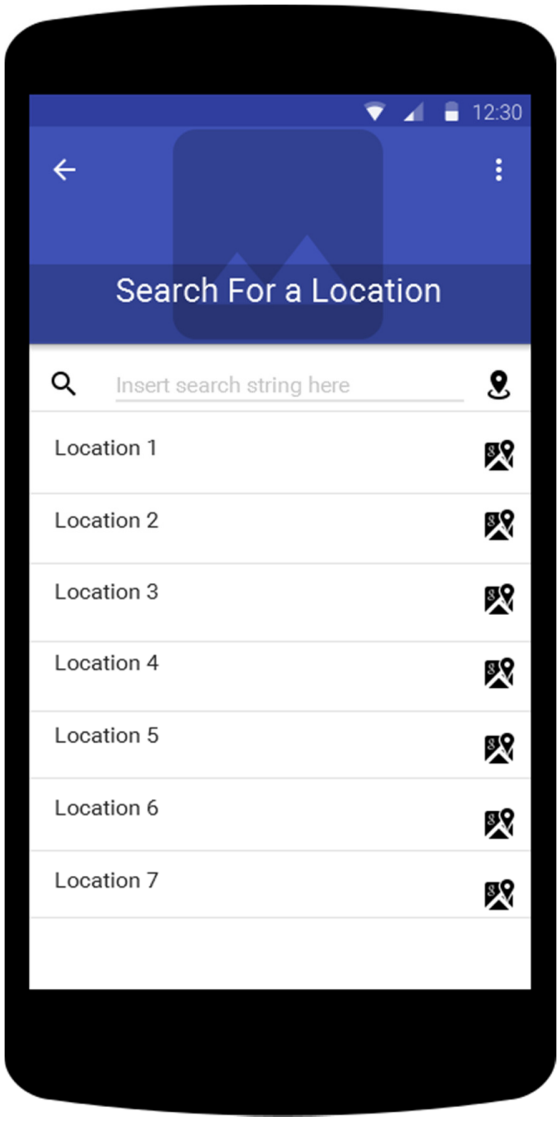
Screen 4



If the user tap the Pen icon on the Action Bar found in the Edit Event screen, it will reach the above screen and he will be able to modify any information by tapping on it.
Most of the editing will be done by a simple Edit Text or Scroll Down list. As for Location information, they could be changed by tiping the name, without having to look for a real location through Google Places. Anyway, if the Google Places is active, the user can tap on the Mapi con and trigger the proper screen which will allow to find the right place thanks to Google Places.

In this page, the Collapsable App Bar will be filled with the image selected by the user.

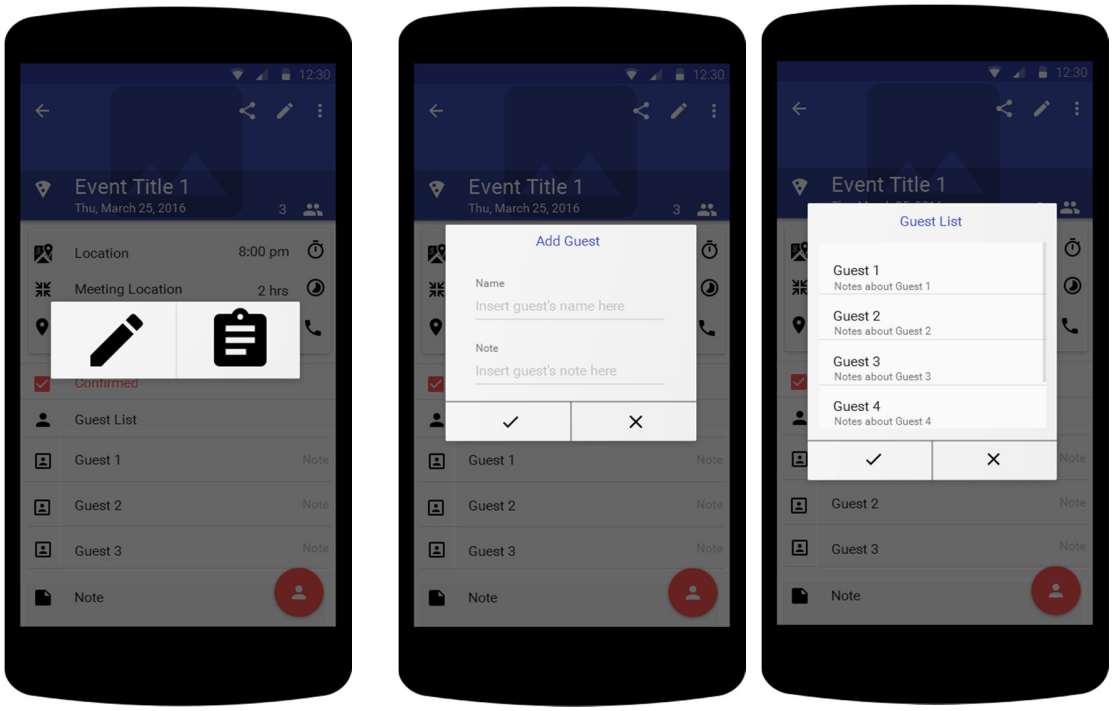
Screen 5



If Google Places usage is enabled, when tapping on the Mapi con near Location's information within the Event's details screen, the user will be brought to the Location Selection screen. Here, the user will have to insert some keywords to ask Google Places for results about them. Outcomes will be presented by a List View and if the user will tap on the item list, the location will be used to populate the Location Field. Otherwhise the user could chose to tap on the Map Icon to following up with Google Map information about the Location Item or finally it's possible to go back if no results will be satisfying.

Finally, by tapping the Place Icon, a Places Picker will pop up to allow the user to select a place near him.

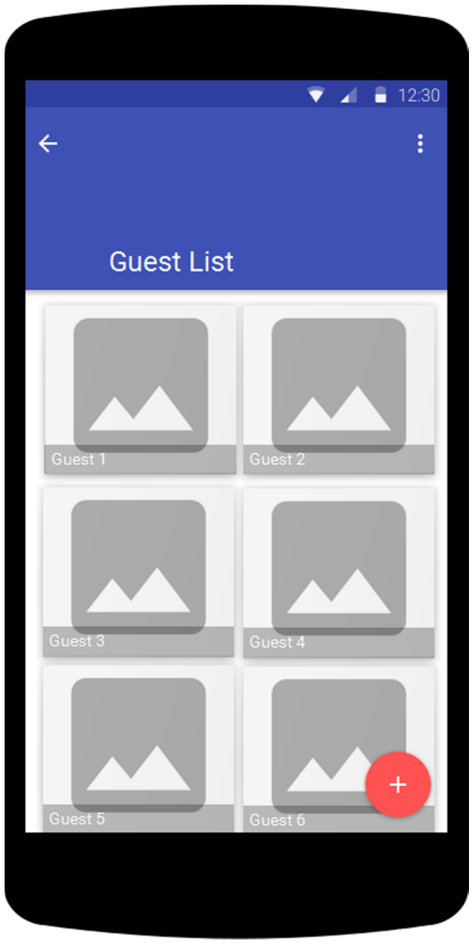
Screen 6



If the user wants to add users to an Event, he will have to tap on the FAB, select the “+” FAB coming out from it and he will have to select how to add a User through a Dialog view:

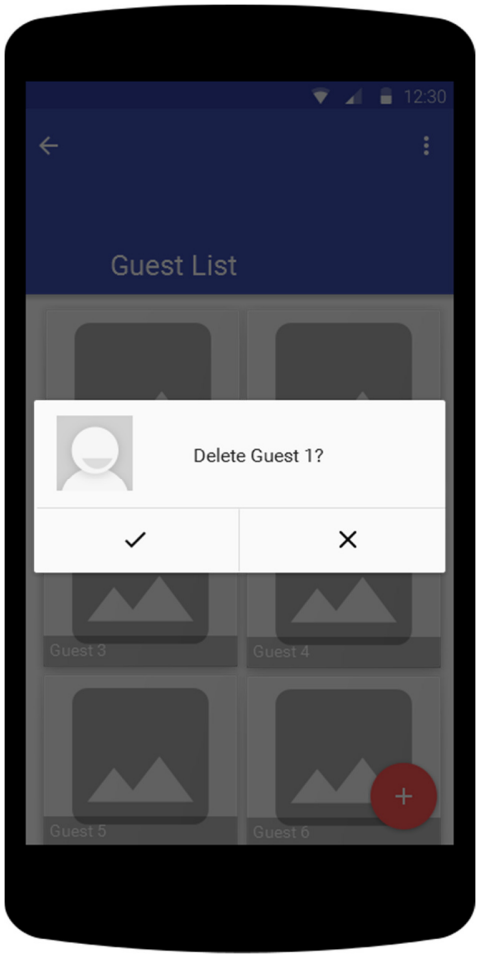
- Manually: the user will have to insert name and notes about that guest.
- From Clipboard: if the user has something understandable in the Clipboard, he will allowed to use it and, if all goes ok, a list of users and their notes will be prompted.

Screen 7



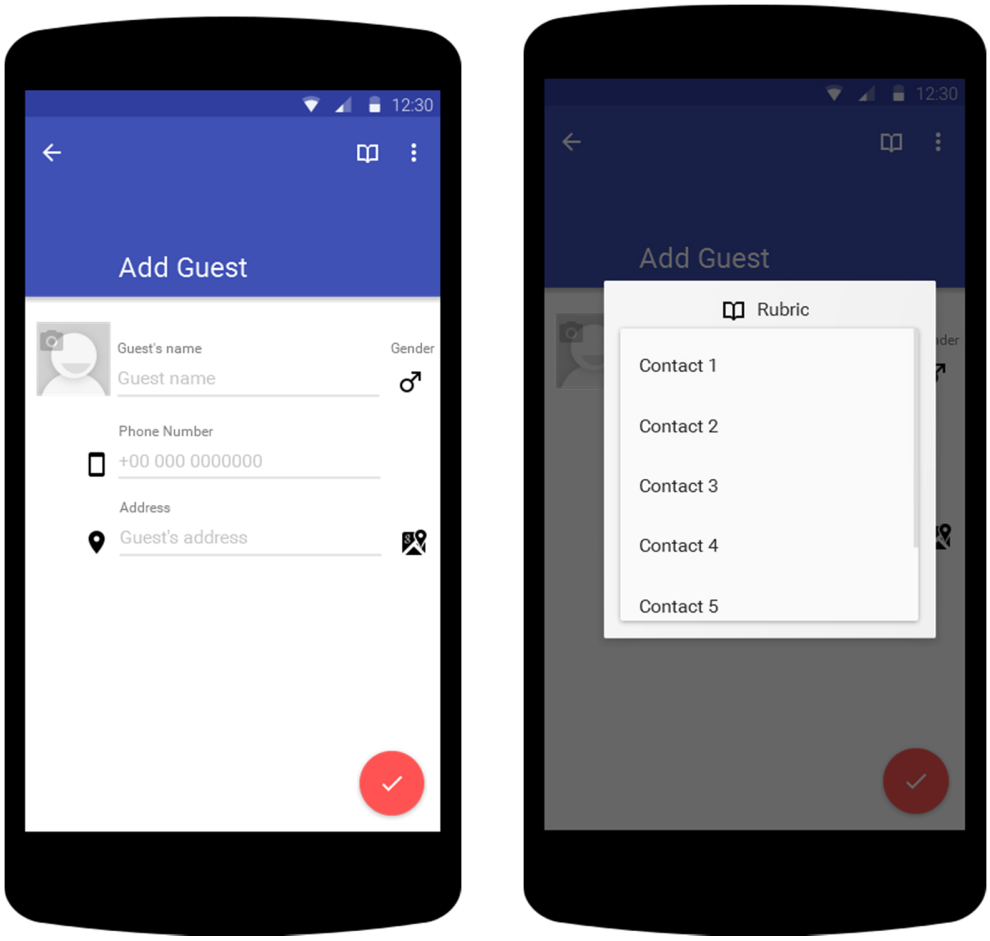
This list is reachable by the “Guests” item of the Overflow menu in the App Bar of the Main Screen. The guests will be presented with Cards.

Screen 8



To delete a guest from the DB, the user has to access to the Guest List screen and hold his finger over the Guest he wants to remove from the DB. A confirmation will be required to succesfully complete the operation.

Screen 9

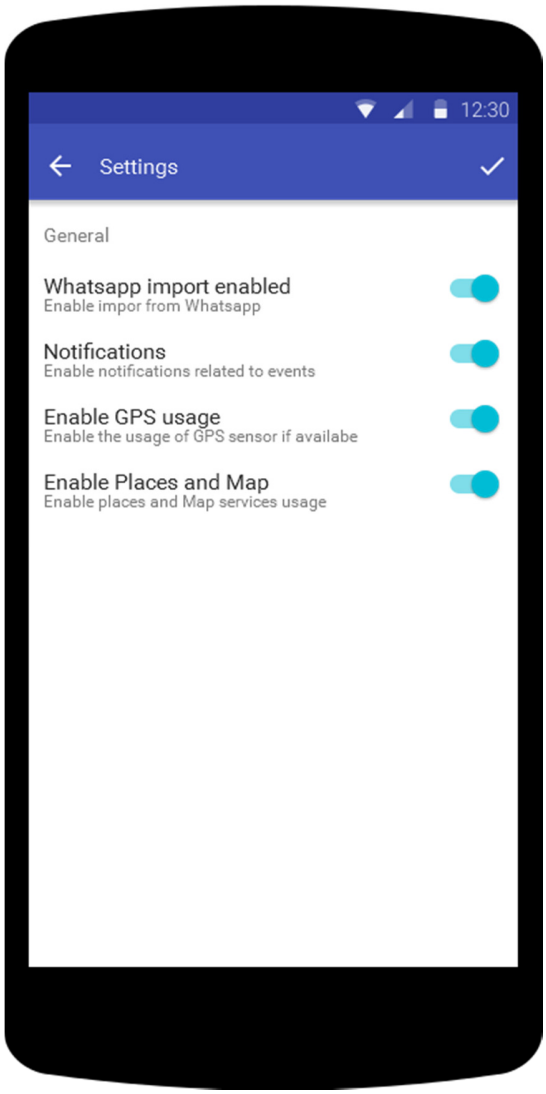


To permanetly add a guest to the DB before creating an Event, the user will have to access to this screen from the Overflow menu item from the Main Screen. Here the user can insert:

- Photo
- Guest name
- Guest gender
- Phone Number
- Address (he can leverage the Map icon to look for the exact address)

The user can also load a Contact information by tapping on the Book Icon in the Action Bar and selecting a contact from the List View .
Tapping on the FAB will save the Guest in the DB.

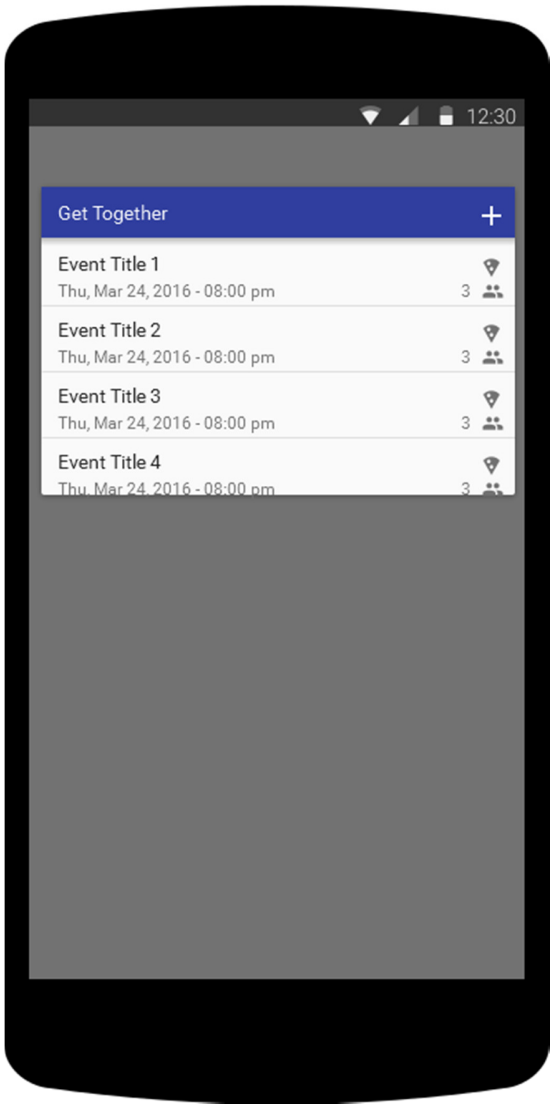
Screen 10



This screen is reachable through the Overflow Icon from the Main Screen.
To save settings the user has to tap on the Check icon in the Action Bar.

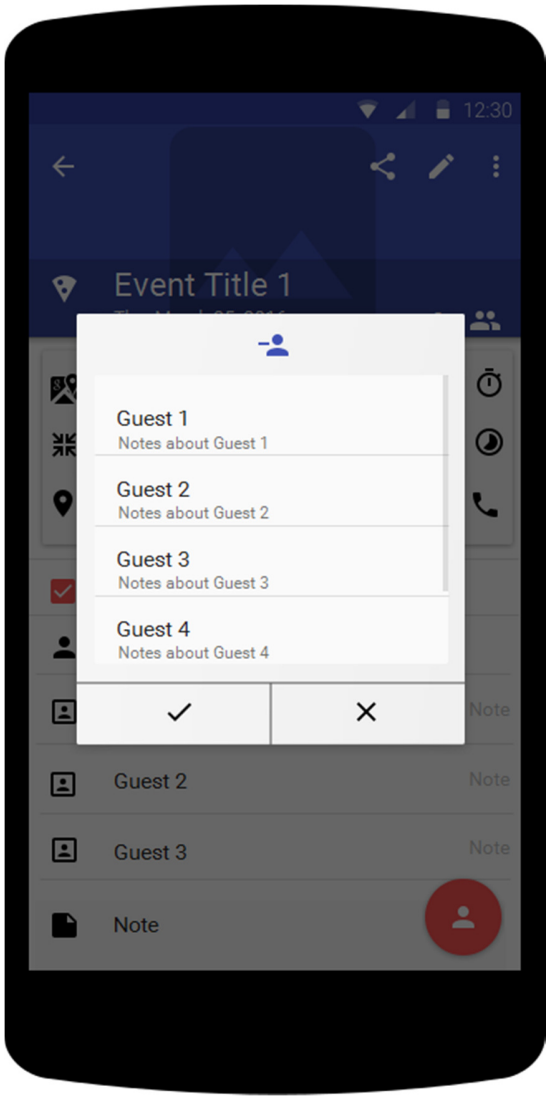
The Whatsapp option enables the capability of parsing the clipboard for relevant data (properly formatted) to manage the Guest List for an event.
The GPS usage will enable the sensor reading and the Enable Places and Map will enable the usage of the Google Services.
Notifications voice will enable notifications about events' time-related information.

Screen 11



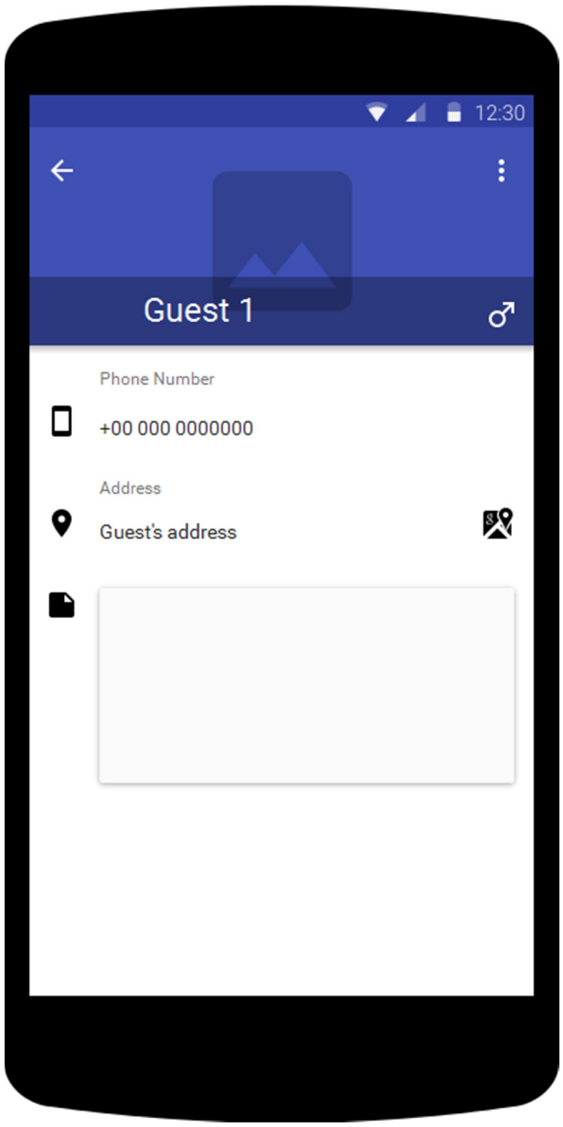
From the widget the user can see the list of the next Events with some related information and he can Tap on the Plus icon to add an event (screen 4).

Screen 12



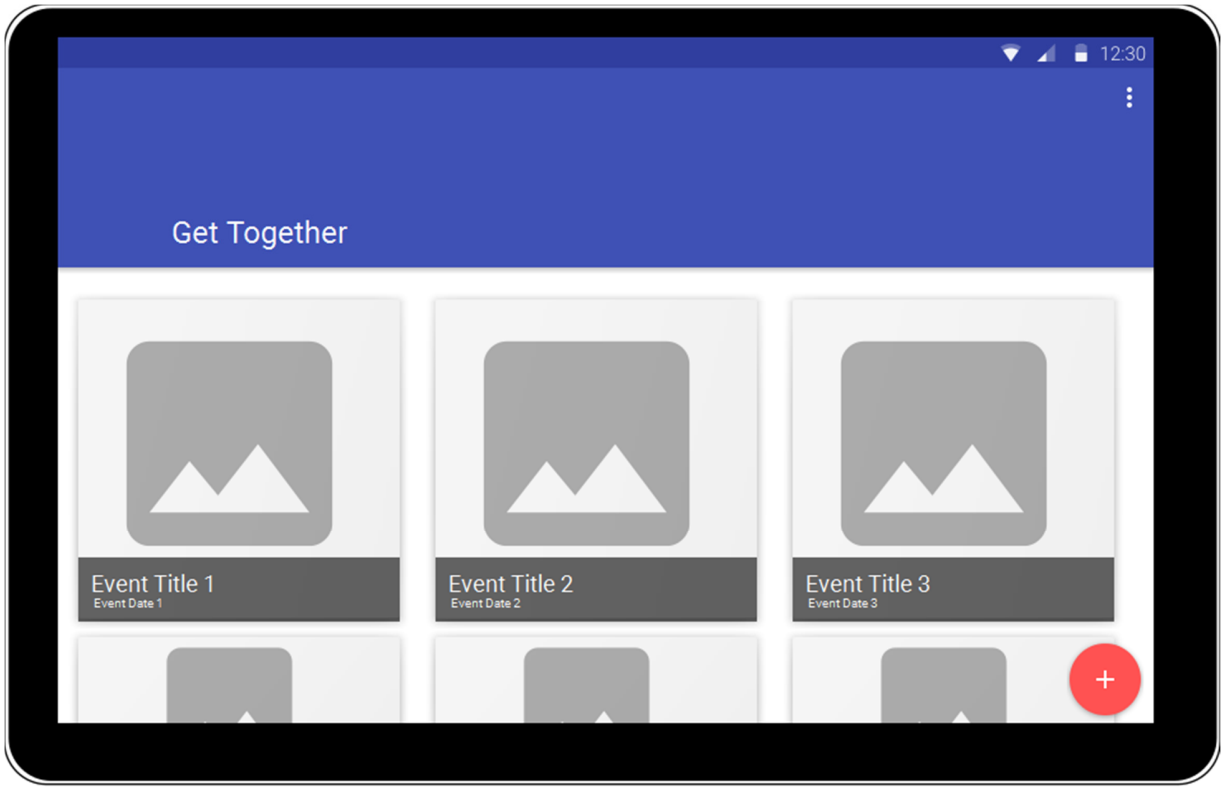
To remove a guest from an event list or from the guest's DB, the user will have to tap on the Minus FAB coming out from the FAB in the Event Details screen or by using the Overflow Icon on the Main Screen.

Screen 13



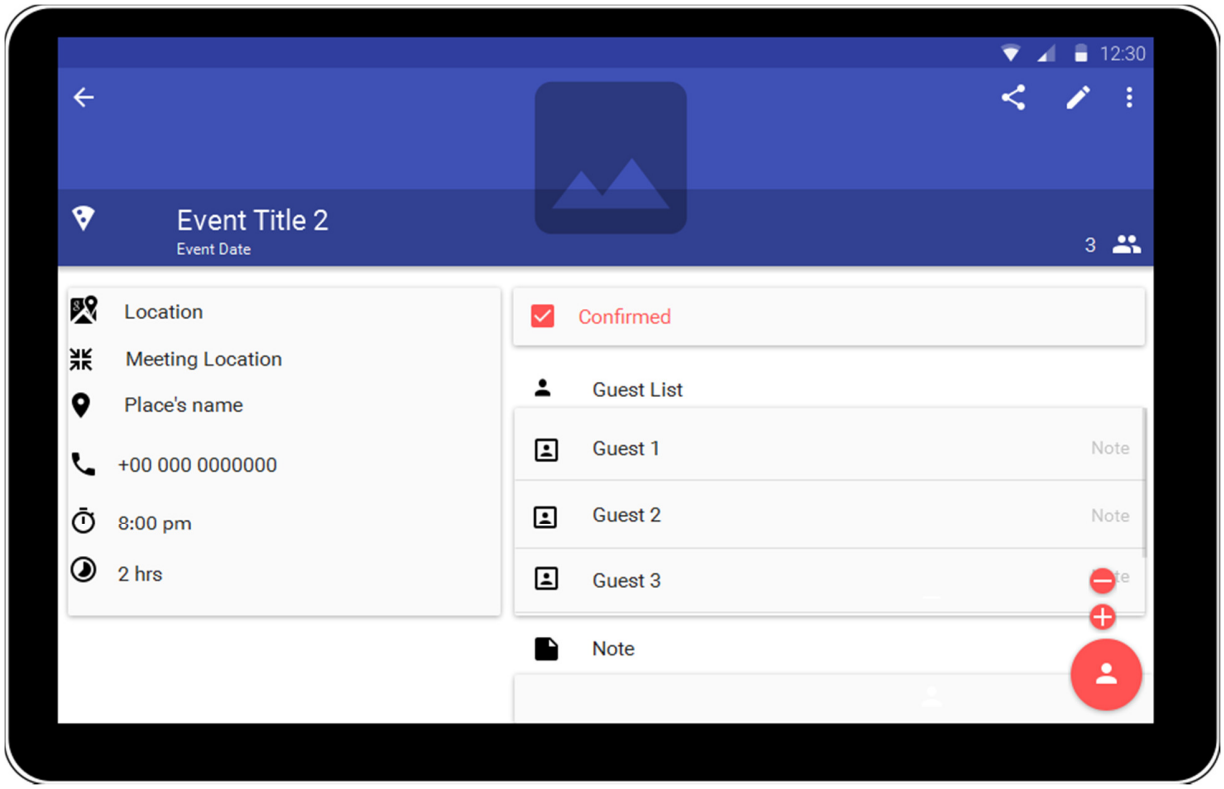
When tapping on a Guest in the Guest List within an Event, the user will see the Guest's Info screen. Some information will be available only if the DB was added to the Guest DB from the Main Screen.

Screen 14



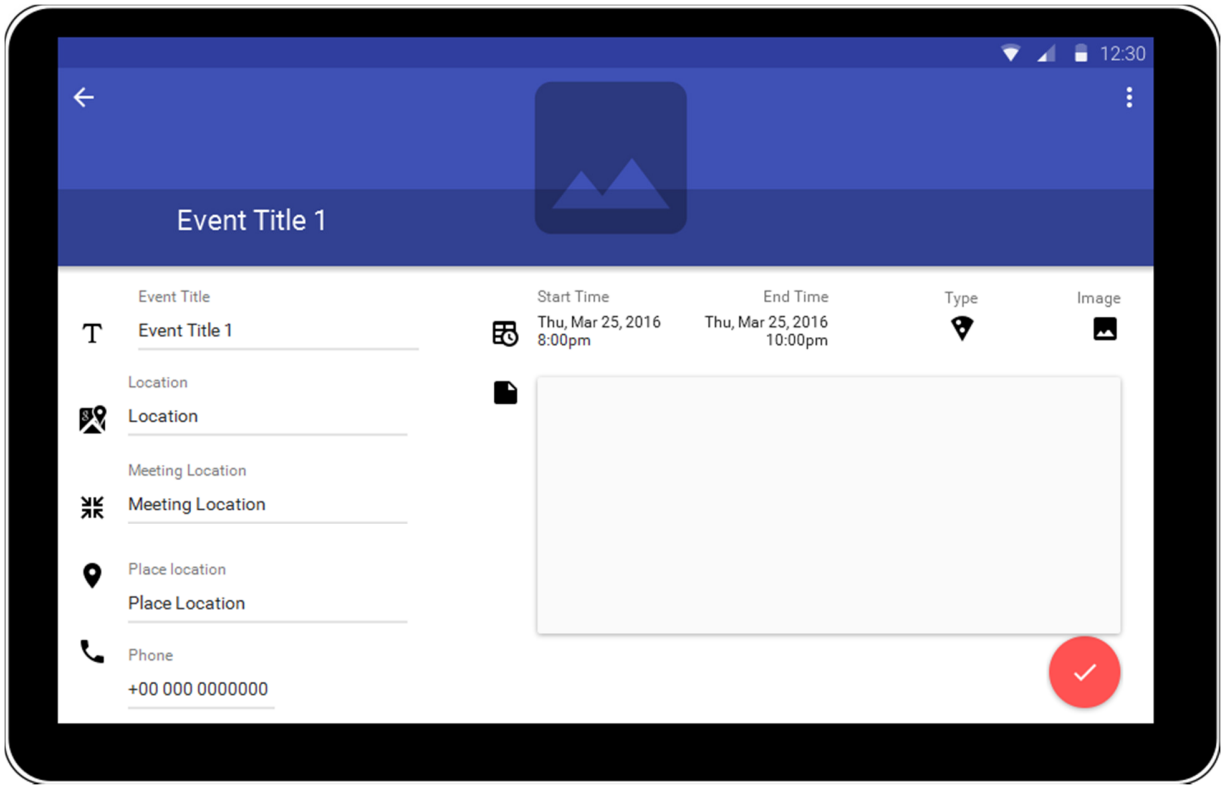
Main screen when the app is running on a Tablet with sw600dp.

Screen 15



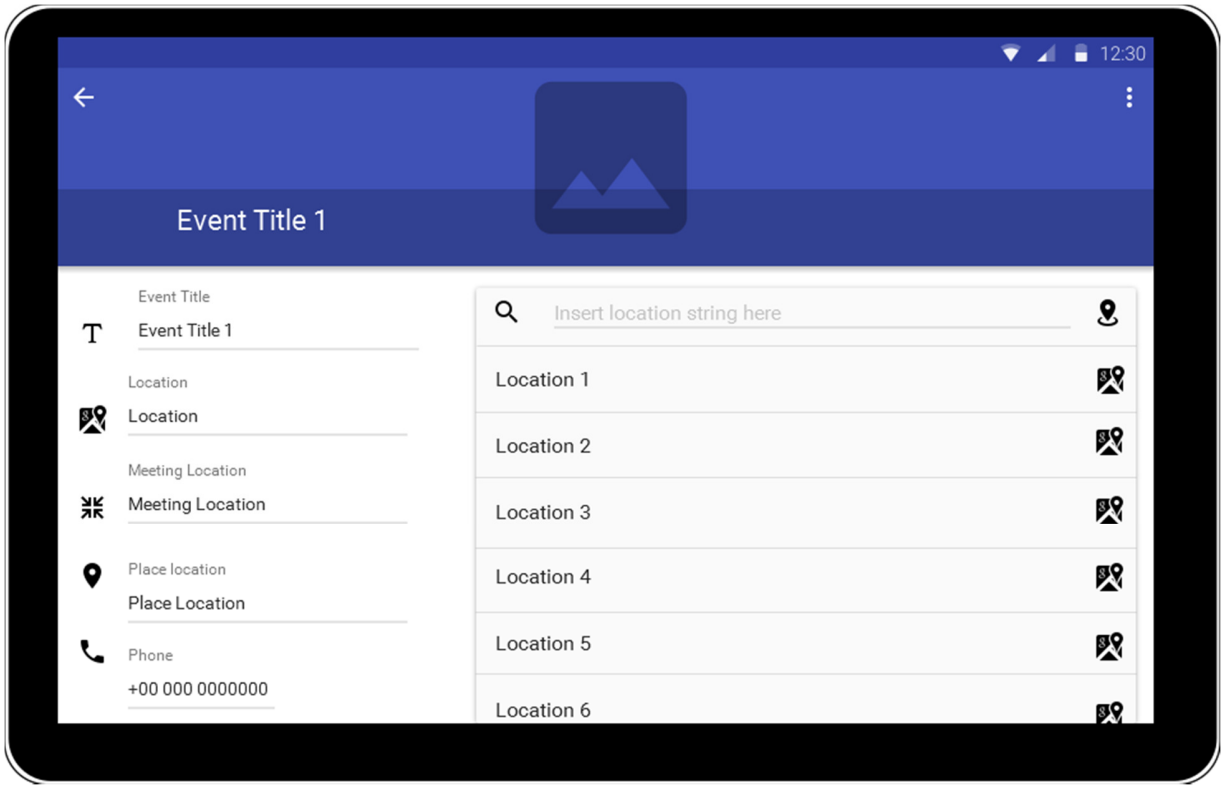
Event’s details screen when in Tablet mode.

Screen 16



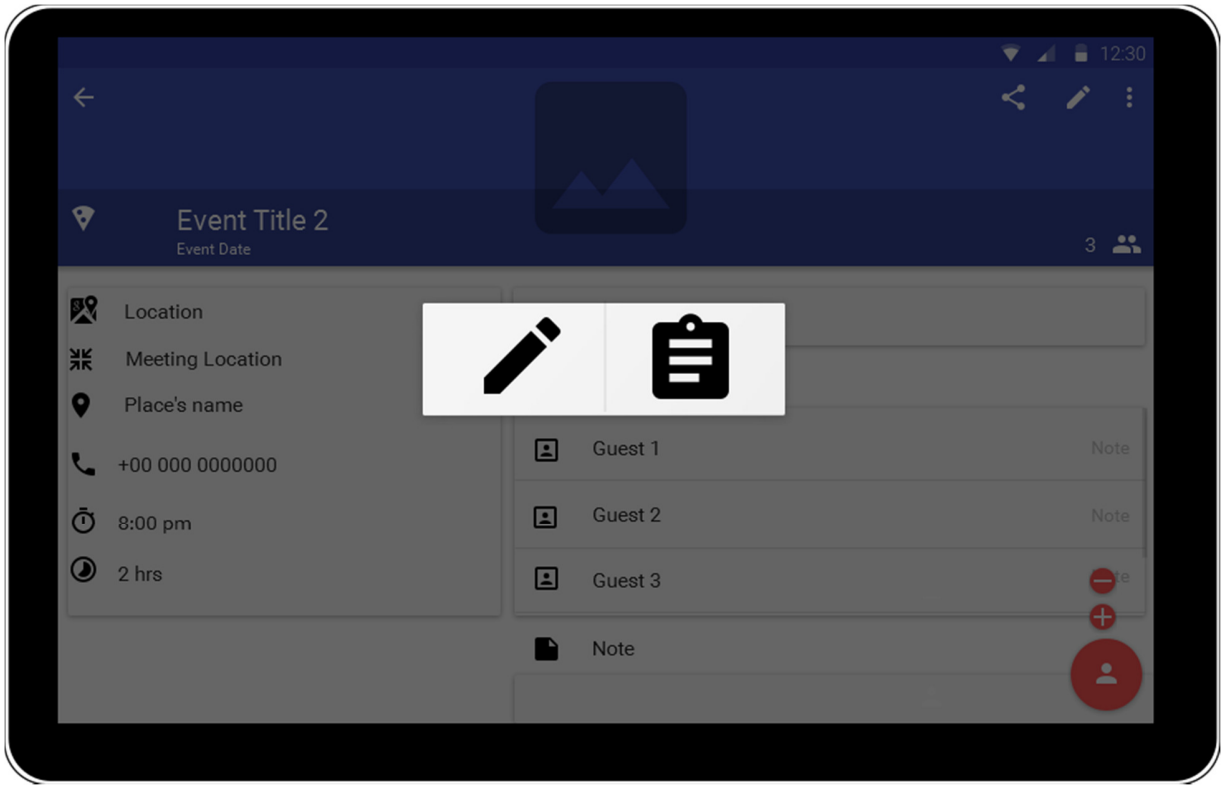
Event edit screen when in Tablet mode.

Screen 17



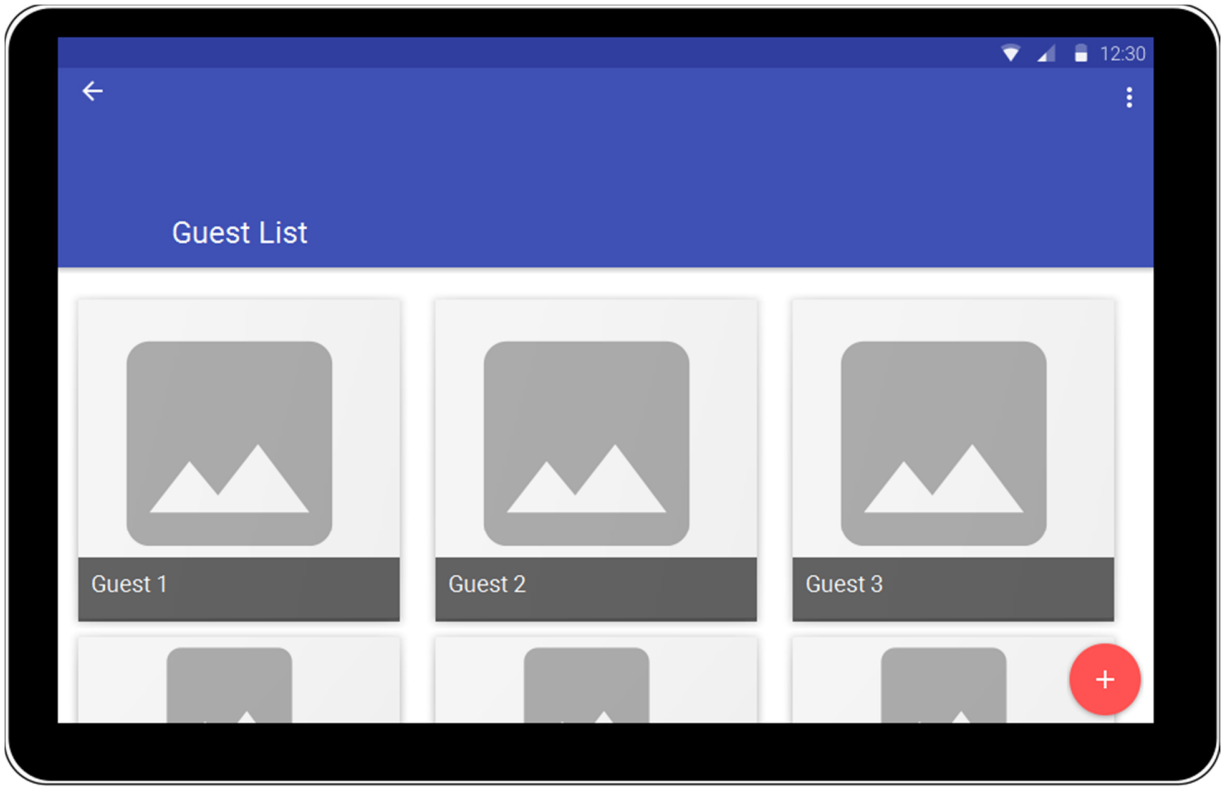
Event location edit screen when in Tablet mode.
The Location Edit will overlay the right half of the Event Details.
By tapping a location result or back button will bring the use back to the main event details screen.

Screen 18



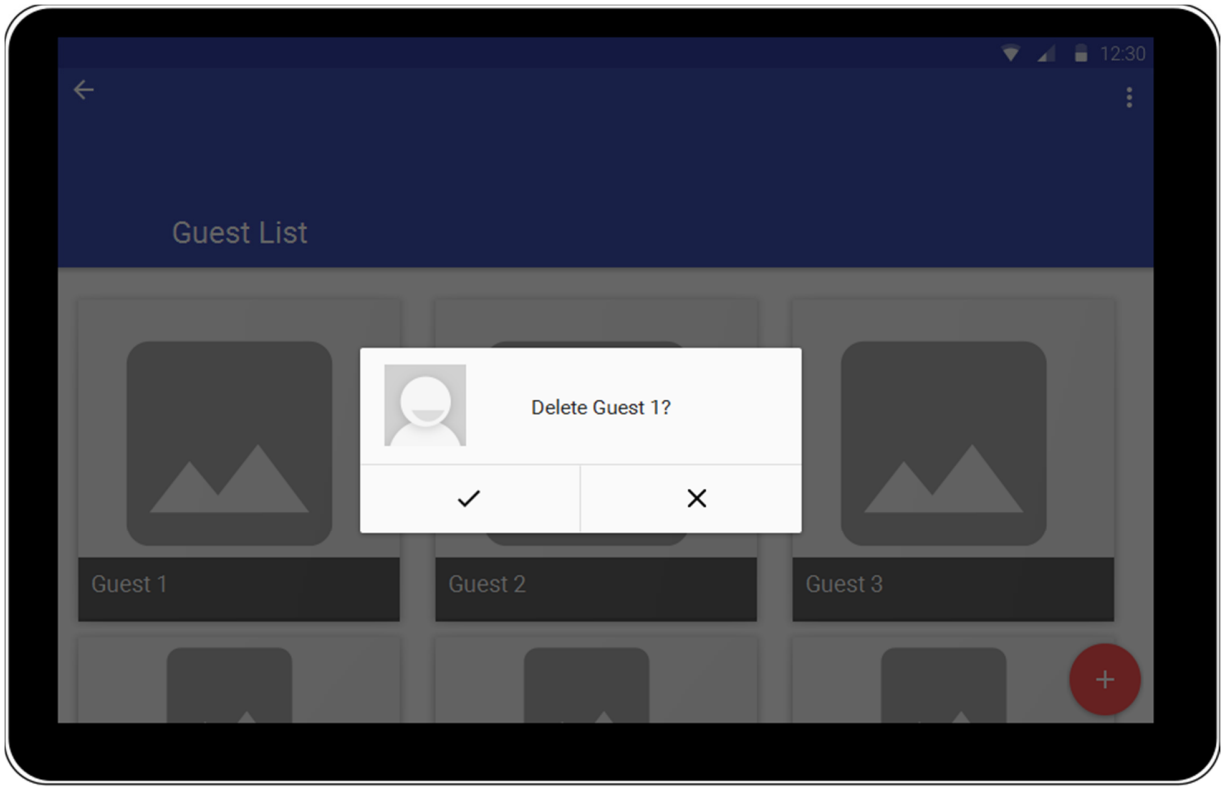
When in Tablet layout, if the user wants to add a Guest to an Event, the same dialog view will be presented like we already seen for Smartphone layout. Dialog view will cover both smartphone and tablet layouts in the same way.

Screen 19



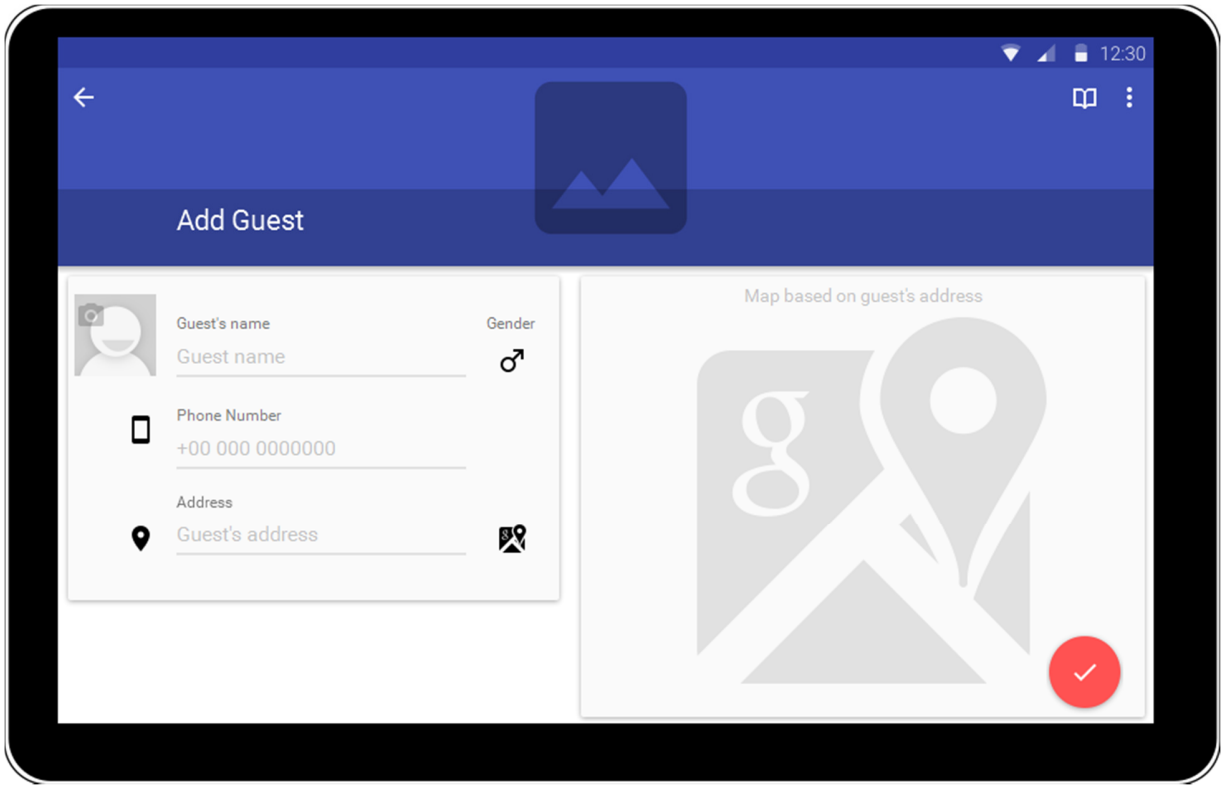
The tablet layout for the Guest List in the DB. From this page the user can Add new users, Delete users from the DB or view information about a particular user by tapping on its card. This screen is reachable from the “Guest” item of the overflow icon in the App Bar.

Screen 20



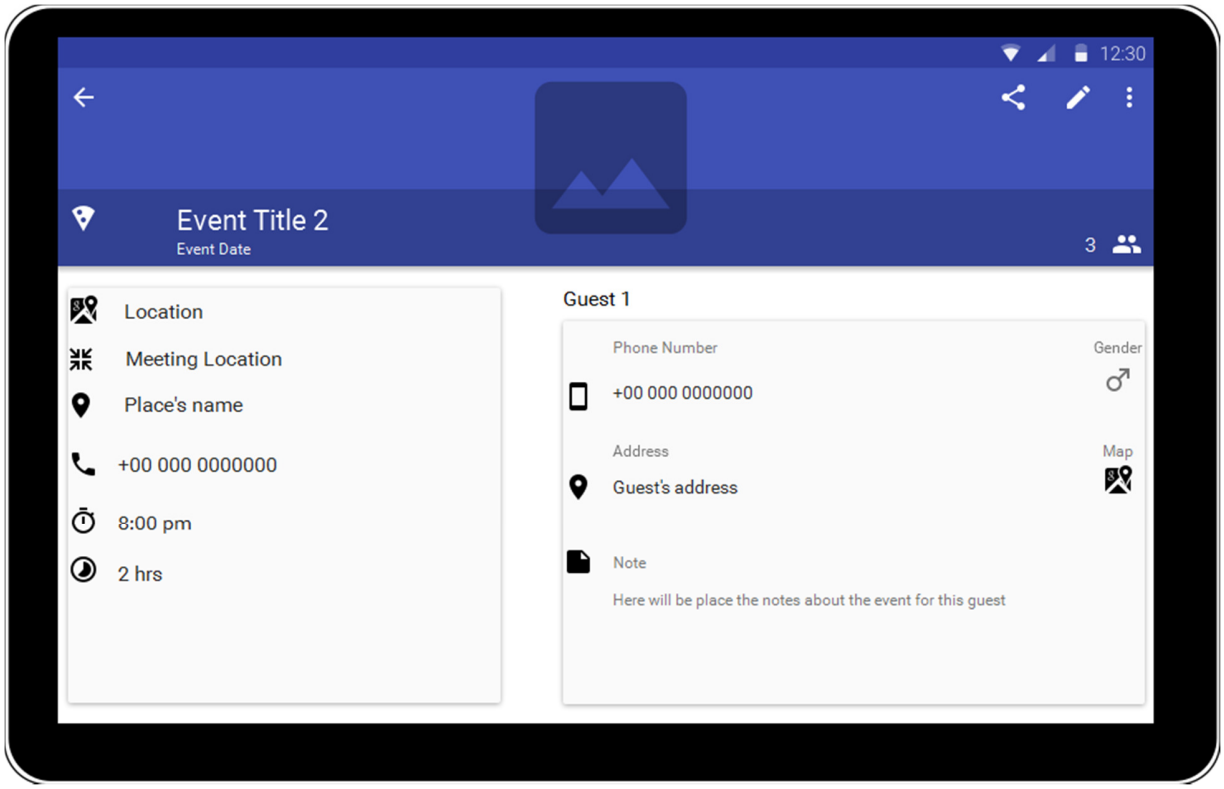
From the Guest List screen, it's possible to delete guests by holding the finger over the Guest's card and then confirm the intention to delete the guest from the DB.

Screen 21



The Tablet Layout provides the above screen when the user wants to add a user to the DB from the main screen. To fill the whole screen, a Map View based on the Guest's address will be available (if enabled).

Screen 22



For the Tablet Layout, when tapping on a Guest within an Event, the app will provide guest's information. If the guest has address information, the Mapi con will be available and i twill bring the user to Google Map by tapping on it.

Key Considerations

How will your app handle data persistence?

The app will store favourite messages with all their attributes and the recently ones into a local Content Provider. Any App settings will be stored with Shared Preferences.

Describe any corner cases in the UX.

- If the user accidentally click the back button when using Google Places service to look for a Location, he will have to go back to the application and the last results will be available on the list

- If the user try to add a user with some required information missing, the app will inform the user about the error found within the form.
- If the user rotates the device while waiting for some results from the Google Places, the system will close and will launch automatically the same request.
- If the user set to use the GPS, but the Location service is not active at OS level, the system will inform the user about that and the function won't be available.
- If the user Add a guest completely equal to another Guest on the DB (except for the Photo), the app won't add the guest and will notify the user.

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

- Glide: used to load every images on the each screen
- Google Play Services Places: used to find places and to use Places Picker
- Google Play Services Map: used to provide Map Views within the app
- Design Support Library: used to enhance the Material Design
- Android Support Library v7: used to leveraging the backward compatibility for some object

Next Steps: Required Tasks

This is the section where you can take the main features of your app (declared above) and decompose them into tangible technical tasks that you can complete incrementally until you have a finished app.

Task 1: Project Setup

- Download, configure and build any library requested by the application
- Define all flavours of the app
- Setup gradle files to correctly import and compile the project with the libraries
- Define activities and fragments for each view and size
- Define the services needed by the application to work smoothly
- Define the animations to be used on transactions
- Define which attributes to include into App Settings, Editing pages and Detail Pages
- Define the permissions needed by the whole application
- Define Clipboard messages format to automatically detect guests and related notes.
- Setup the AndroidManifest.xml file

Task 2: Implement UI for Each Activity and Fragment

- Build UI for the Main Activity
- Build UI for each planned Activity/Fragment reachable from the Main Activity
- Implement Animation and shared elements with icons and fake images to setup the behavior
- Implement UI for Tablet versions

Task 3: Create the Guests Content Provider

Creation of the Content Provider and Content Resolver for the Guests.

- Define the Table Structure with:
    - Name
    - Gender
    - Address
    - Photo
    - Phone (_ID) required
- Define Content Provider and Content Resolver
- Implement Contract and Helper methods
- Implement loaders

Task 4: Create the Events Content Provider

Creation of the Content Provider and Content Resolver for the Events.

- Define the Table Structure with:
    - Event title
    - Image
    - Event Type
    - Event Start Time
    - Event End Time
    - Event Location
    - Event Meeting Location
    - Event Place Location
    - Event Phone Number
    - Event Note
    - Event Guest List as BLOB type
- Implement Serializer/Deserializer for the Event Guest List column
- Define Content Provider and Content Resolver

- Implement Contract and Helper methods
- Implement loaders

Task 5: Implement Google Play Services - Map

Implementation of the Map View feature which allows the user to navigate within a Map starting from a given address.

- Implement the connection with Google service
- Handle click events on each Map Icons for the whole App
- Manage the Map View for each view designed to contain it.

Task 6: Implement Google Play Services - Places

Implementation of the Google Places service to leverage the Places Picker view and the places search by strings.

- Implement the connection with the Google service
- Handle click events on the Place Icons
- Handle search by string through text views
- Implement Places Picker view when tapping on Place Icons
- Handle callbacks for Places results

Task 7: Implement Whatsapp parser

Implement a parser for text coming from Clipboard for messages copied from Whatsapp's chat.

- Define a standard format for multiple messages copied from Whatsapp's chat.
- Define a standard behavior when detecting something in the Clipboard.
- Define a custom format for single and multiple messages to automatically import information in the event from the Clipboard
- Implement the parser
- Implement a results which detect with kind of message has detected (Whatsapp, whatever)
- Implement the filler (from data extracted by the parser)
- Define a utility for this functionality which could work along with some other future parsers.

Task 8: Implement the Widget

Implement the widget to properly show next events in a list and to add an event directly from the Home screen.

- Create a Widget to communicate with the app
- Show next events information within a List View in the Widget
- Handle click event for each item of the List View to access directly to that event
- Add a Plus button to access to the screen to Create an Event

Task 9: Handle Error Cases

Test and handle error cases related to:

- Network communication failures such as Timeout, Server errors, Network not available
- Data errors received from the Servers.
- Errors on User's inputs when inserting empty or unaccepted values on required fields
- Content Provider operation errors

Task 10: Produce signed .APK

Generate the signed version of the APK to deploy