GitHub Username: kkho

SocialMe

Description

Based on where you are. And if you are traveling alone like myself. You want to know what event that country you are traveling to have. You can also check out events around your place too.

Intended User

Friends, Travelers, families and people that are interested in traveling or social people

Features

List the main features of your app. For example:

- Saves information
- Check Events on your country and other places
- Save information of event you are going

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

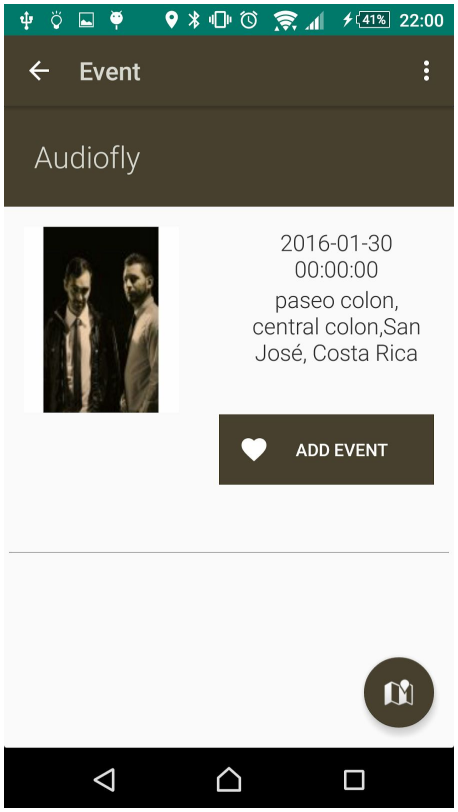
Replace the above image with your own mock [click on the above image, then navigate to Insert → Image…]

During the homescreen, we will get a lit of events which is listed for this week. It is then filtered based on things you want to do when you are on vacation (art, musical, festival, concerts, technology etc). When clicking an item, you get to a more details about the event. The event will need to know the GPS location of where you are now and also need to have internet.

Some don't have images so I just use a default image placeholder just in case they don't have it.

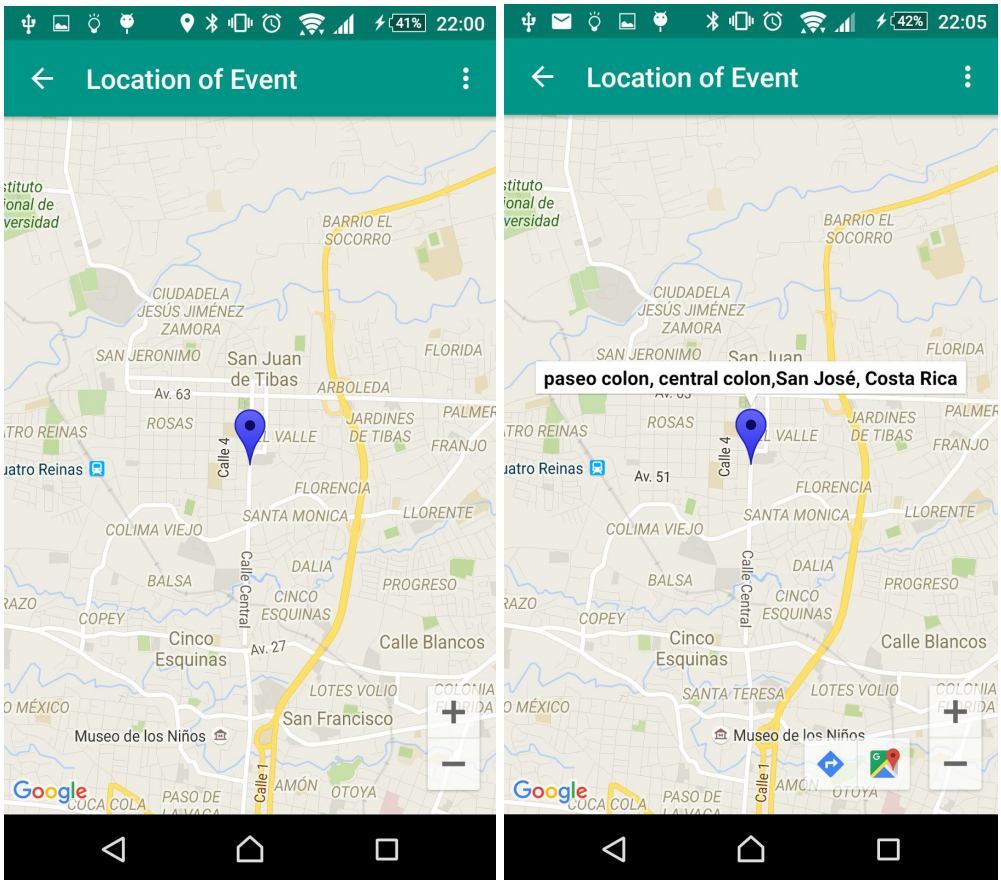
There will also be facebook log in so that image, and information is stored into the stored user.

Screen 2



On each event item, you will see a detailed information about it and also share, add the event and also you can find the location of the event which will be marked in the location of where it is on the map. To do that, just press the Floating Action Button which will lead you to Screen 3

Screen 3



When pressing on the “SHOW ON MAP” button, you will then get to where the location is at the specific address with the correct marker on the map. Pressing on the marker gives the specific address of the location.

Key Considerations

How will your app handle data persistence?

I will use ContentProvider to store previous searched items and favourites should be stored as a HashSet for ids. The Id HashSet will be stored using ContentProvider.

Describe any corner cases in the UX.

You will return to the main screen when you are in another Activity. If you have done an operation. And hit either a “Done” button or the back button.
When you swipe, you will refresh new data and delete whatever contentprovider has stored.

What will be challenging is image transparency using Android Material Design principle.

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

Picasso to handle the loading and caching of images.
ButterKnife for easily binding components form layout to code
GSON for easy Json serializing and deserializing
CircleImageView that was created by hdodenhof for the navigationview. This is to store the facebook profile image.
Facebook SDK This is used to get information from the user and use the profile image etc.

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

- Configure libraries
- Add resource files and icons
- Create Facebook Login
- Create Calendar implementation (store it locally and I will not use Google Calendar)
- Put API key in a secure place and make sure not to commit it. I will use an api from : https://api.eventful.com/

Task 2: Backend Implementation

- Setup necessary Objects to store in Local Database
- Create webservices to do the necessary calls to store information
- Load images using Picasso when creating RecyclerView Adapters.

Task 3: Implement UI for Each Activity and Fragment

- Build UI for BaseActivity, MainActivity
- Build UI for Fragments
- Implement Share button to share for example on facebook, etc This will be implemented into each event items.

Task 4: Bug and Feature fixing

- Implement Google play Services (Google maps, Geolocation)
- Add gradle.properties to fill in values so it can be generated rather than hardcode all values inside gradle files (version code, application name, api keys, etc).
- Handle network errors, UI errors, etc