Capstone Stage 1

GitHub Username: FelixGruener

urLoco App

Description

Ever seen a wonderful Place on a Picture and wondered where that place exactly is ? Ever lost and wondered where you exactly at ?

With my App you can simply take a picture and it will classify your location with the Name of the Place and important Labels to identify the Place . Clicking on Details will give you exact GPS location of that Place shown on Google Maps together with Options how to get there. This App will be written solely in Java.

Intended User

- Travel Enthusiasts.
- Any User who got lost.
- Any Person who want to play a travel or intends to go to a special Place

Features

- Read, create, update , delete a location classification
- Pick a existing Picture from Gallery
- Takes pictures
- Uploads Picture
- Shows Response from Location Classification Service
- Shows Details of the Location Classification (e.g. GPS Position in Google Maps).

User Interface Mocks

Main Screen

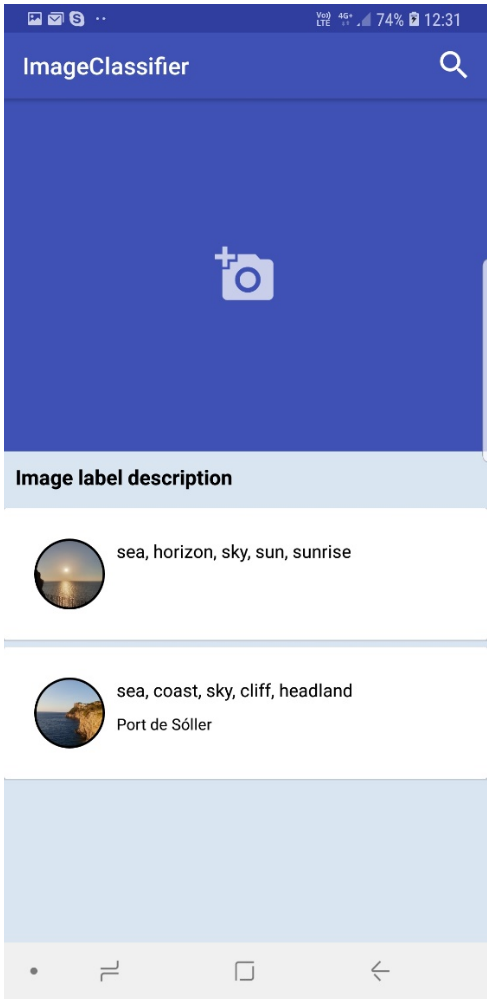


Image label description

sea, horizon, sky, sun, sunrise

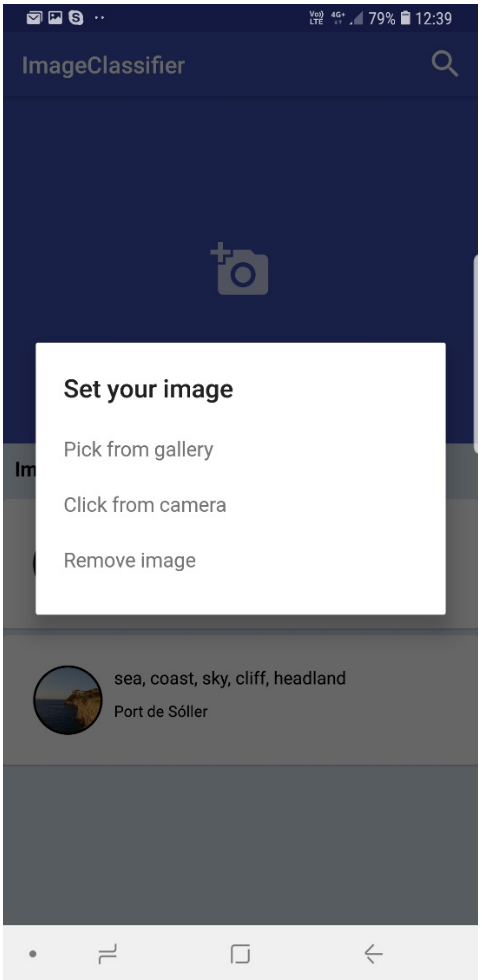
sea, coast, sky, cliff, headland

Port de Sóller

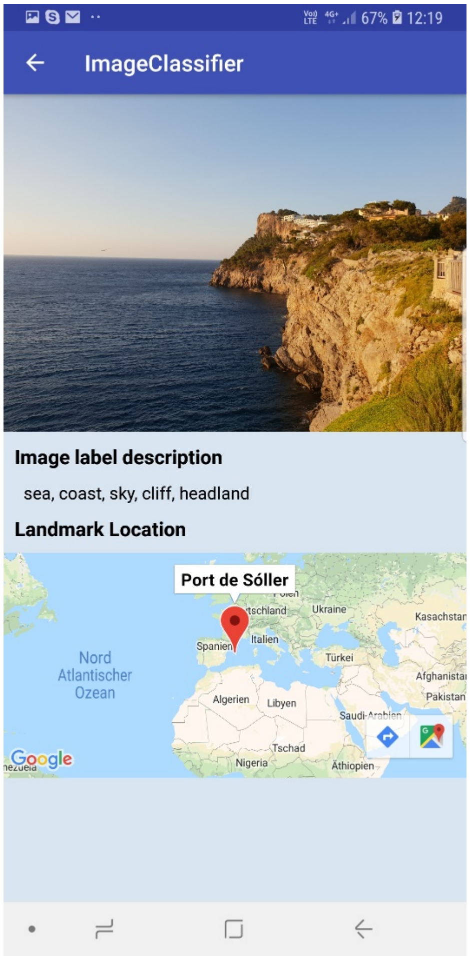
User can click on Camera Icon in upper middle of Screen to choose from several Options:

- Pick from Gallery
- Take new Picture
- Delete current Picture

Mock Choose Option:



Detail Screen with located Place and Google Maps Service



ImageClassifier

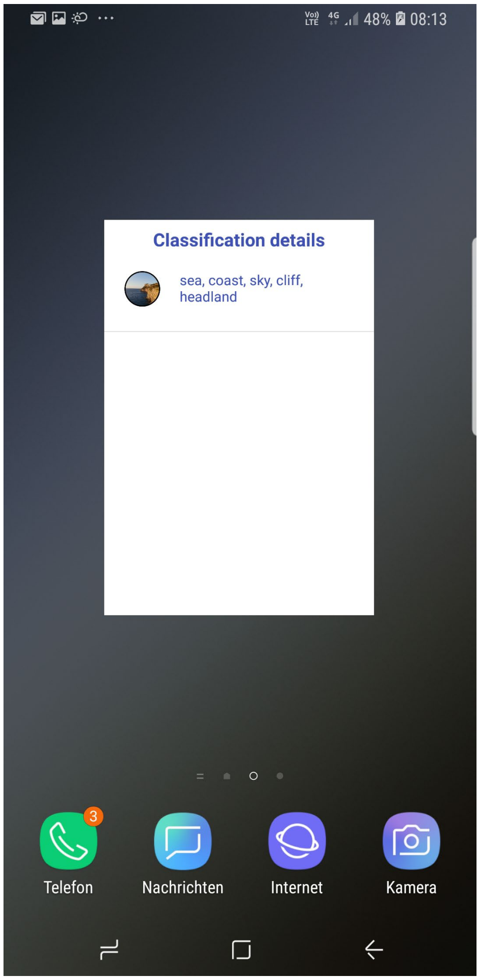
Image label description

sea, coast, sky, cliff, headland

Landmark Location

= > The User gets the located Place Info/Details shown and the Option to switch to Google Maps to use further Routing Options.

Widget listing the latest Entries:



Classification details

sea, coast, sky, cliff, headland

Key Considerations

How will your app handle data persistence?

Saved State for switching and flipping between Portrait and Landscape View . Classification History will be persistent in SQLite Database which along as Content Provider. A response body containing the classified meta data of the image is saved locally into the SQLite database. Records can also be deleted

Describe any edge or corner cases in the UX.

- Unstable or missed network connection: the application must not crash in that cases
- Device orientation change: the application must handle all long running operations correctly considering possible configuration changes
- UI freezes: the application must not use the main thread for any resource consuming operations

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

Naming here only the important ones:

Retrofit 2 (Version 2.3.0): for network API requests, a type-safe HTTP client for Android and Java
Butter Knife (Version 8.8.1): for boilerplate code reducing, it helps in view binding
Glide (Version 4.8.0): for images loading. Glide is a fast and efficient image loading library for Android focused on smooth scrolling. Glide offers an easy to use API, a performant and extensible resource decoding pipeline and automatic resource pooling.

Also will be using :

Android Studio 3.1.4
Gradle Version 4.4
Gradle Plugin Version 3.1.4

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

Google Vision API : classifying different uploaded images
Google Maps: for the displaying location of uploaded images

In this project I utilize one google play service which is the google map and google vision API using image classifier. After user takes snap shot from camera or select image from the gallery, it uploads the image file to the Google image classifier server for classification using https://vision.googleapis.com/v1/images:annotate

Next Steps: Required Tasks

Task 1: Project Setup

1. Create Default android project with main view.
2. Add this proposal and and additional feedback from Udacity advisor(s).
3. Add ReadME page.
4. Create App Structure Folders
5. Implement Dependencies

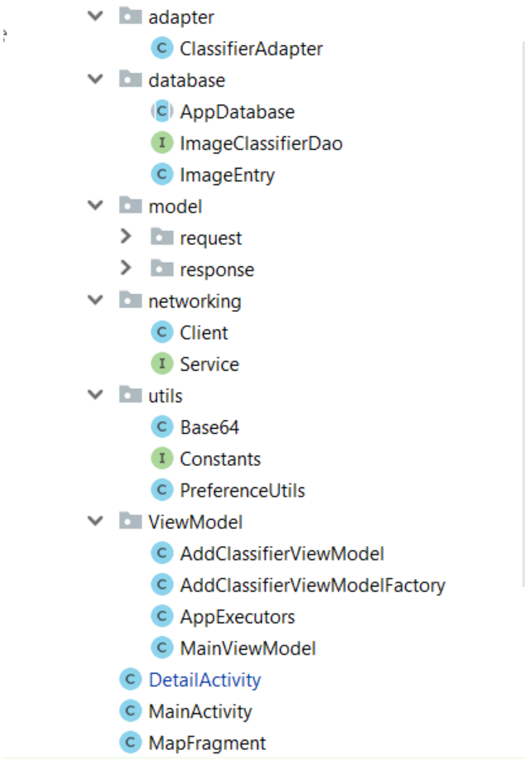
Task 2: Implement UI for Each Activity

List the subtasks. For example:

- Build UI for Main Activity
- Build UI for Detail Activity
- Build UI Options Menu

Task 3: Implement Modules

Mock of App Structure and Classes:



```
adapter
    ClassifierAdapter
database
    AppDatabase
    ImageClassifierDao
    ImageEntry
model
    request
    response
networking
    Client
    Service
utils
    Base64
    Constants
    PreferenceUtils
ViewModel
    AddClassifierViewModel
    AddClassifierViewModelFactory
    AppExecutors
    MainViewModel
DetailActivity
MainActivity
MapFragment
```

Implement Services:
In this project I utilize one google play service which is the google map and google vision API using image classifier. After user takes snap shot from camera or select image from the gallery, it uploads the image file to the Google image classifier server for classification using https://vision.googleapis.com/v1/images:annotate

The Google play service google maps, is used to identify landmark locations from the image i.e using latitude and longitude coordinates to determine some landmark destination detected in the image classified. API key from google developer console before working google maps is needed. Using Intent Service.

Implementing the Widget. Setting up the following

- AppWidgetProviderInfo object:
  Describes the metadata for an App Widget, such as the App Widget's layout, update frequency, and the
- AppWidgetProvider class. This should be defined in XML.
  AppWidgetProvider class implementation
  Defines the basic methods that allow you to programmatically interface with the App Widget, based on broadcast events. Through it, you will receive broadcasts when the App Widget is updated, enabled, disabled and deleted.
- View layout
  Defines the initial layout for the App Widget, defined in XML.

Resources such as Images are only referenced in DB and not stored directly in the DB.

Task 4: Implement Error Handling

For all Eventualities Error Handling needs to be implemented:

- Unstable or missed network connection: the application must not crash in that cases
- Device orientation change: the application must handle all long running operations correctly considering possible configuration changes
- UI freezes: the application must not use the main thread for any resource consuming operations

Task 5: Test the App

App must be tested on several Devices such as Tablets , smartPhones (on Emulator aswell real Devices).