
GitHub Username: hyewon0723

Travelogy

Description

Google Photos has a place feature under Album. It somewhat groups the photos by geographic area. This application organize your Photo by Country you have been visited and view your traces on the map.

Intended User

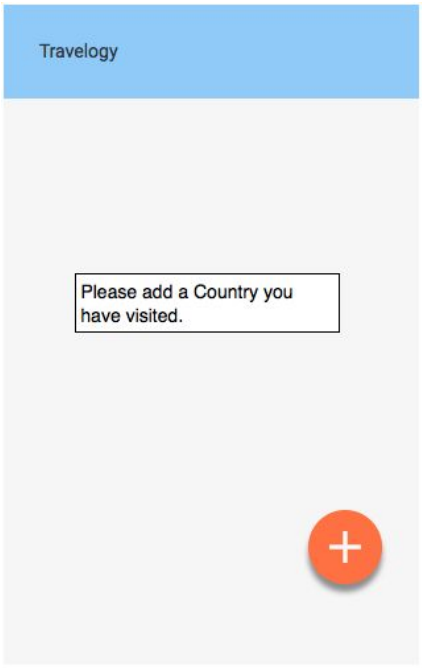
User who loves to travel around the world and want to keep track of which country they have been visited and see their trace where they have visited.

Features

- Keep the record of countries you have visited
- Organize your pictures by countries.
- View places you have been to in google maps

User Interface Mocks

Screen 1

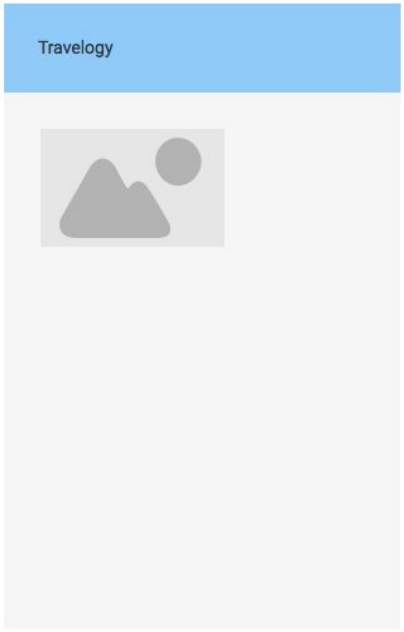


This is the first screen that user will see. This screen will have instruction text. User will select floating button to add country they visited.

Screen 2

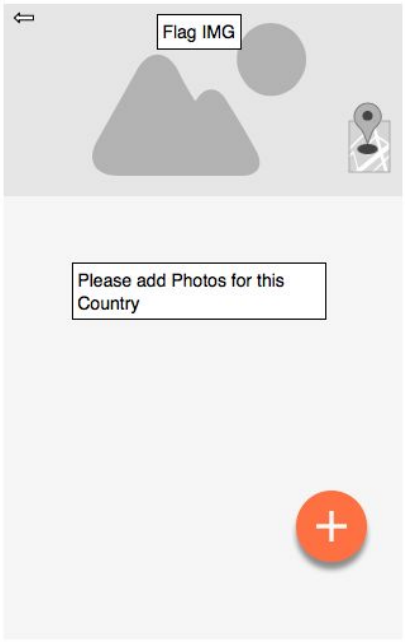
User will enter name of country they visited. The application will search the country code in the background and prompt the message if not found.

Screen 3



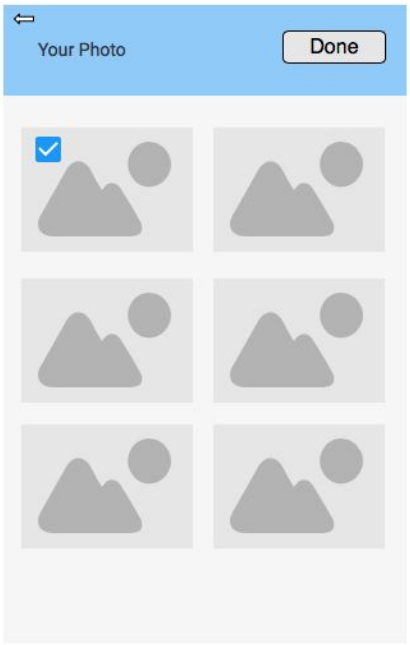
If found, the app will retrieve the country flag image using country code and display the flag image on the main screen. When user select the flag image, it will transit to photo viewer screen.

Screen 4



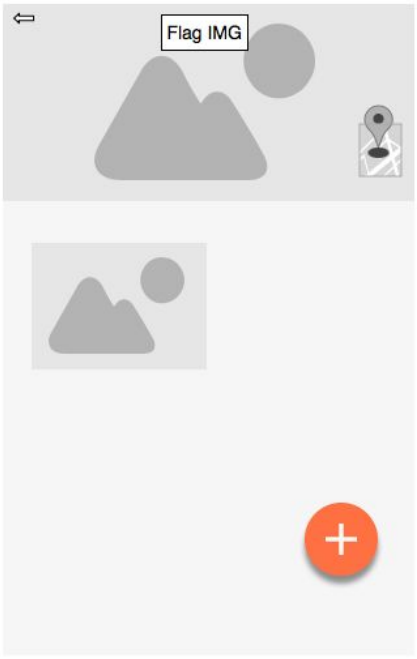
This screen will have instruction text to add photo that was taken in corresponding country. User will select floating button to add photo.

Screen 5



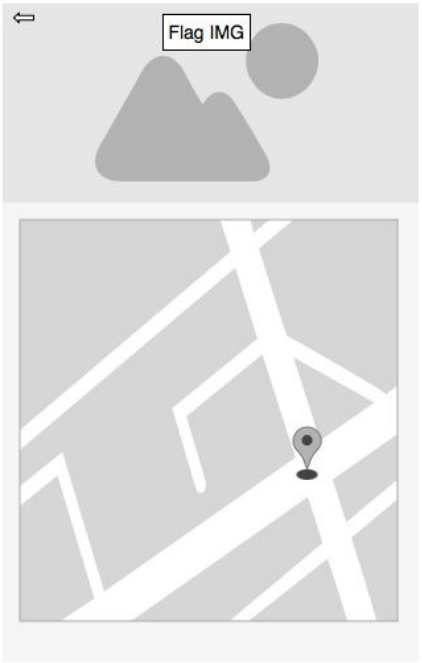
When User will select floating button to add photo, the application will load the photo library from mobile phone. Then user will select the photo from photo library.

Screen 6



The select the photo will display on the screen.

Screen 7



When user selects the map icon, it will display the location of photos that were taken in the country on the map.

Key Considerations

How will your app handle data persistence?

Will build content provider to hold data (country code, location data, image file of local path) on the device using SQLite.

Describe any corner cases in the UX.

When user enters wrong country name, app will prompt the message that they entered wrong country name. If uploaded image doesn't have location data in exif and user selected the map icon, app app will prompt the message that location data is not found in the photo.

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

Will use Picasso or Glide to handle the loading and caching of images. Use SQLite for local data storage

Describe how you will implement Google Play Services.

Will use google maps to display locations that photo was taken and implement ASyncTask to retrieve the country code of user input from https://restcountries.eu and pull the flag image from http://www.geognos.com/geo/en/world-countries-API.html

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

- Create new project
- Configure libraries and add dependencies.

Task 2: Implement UI for Each Activity and Fragment

- Build UI for MainActivity that will query the database and display the list of country flag in tile view.
    - MainActivity will have FAB for adding a new country

- Build UI for DetailActivity that will be launched by MainActivity and display the photo in tile view.
    - DetailActivity will have FAB for adding a photo from local storage
    - UI will display Flag image on the top and map button
- Build UI for MapActivity that will be launched by DetailActivity and display the location that picture was taken.

Task 3: Implement SQLiteDB

- Create DB Statement
- Insert into DB statement
- Query DB and return results statements
- Interage results with tile view through adapter on main and detail activity

Task 4: Tablet View

- Add alternative layouts
- Integrate with mobile view UI flow

Task 5: Testing

- Create test case and run the case. Go over the project rubric and see if there is anything missing.

Submission Instructions

1. After you've completed all the sections, download this document as a PDF [File → Download as PDF]
2. Create a new GitHub repo for the capstone. Name it "**Capstone Project**"
3. Add this document to your repo. Make sure it's named "**Capstone_Stage1.pdf**"