

Mobile Challenge - Ualá

This is a single challenge that may be resolved in Kotlin for Android, or Swift for iOS.

Goal

The goal of this assignment is to evaluate the problem solving skills, UX judgement and code quality. Consider how you would approach it on a production level when building it.

Requirements

We have a list of cities containing around 200k entries in JSON format. Each entry contains the following information:

```
{  
  "country": "UA",  
  "name": "Hurzuf",  
  "_id": 707860,  
  "coord": {  
    "lon": 34.283333,  
    "lat": 44.549999  
  }  
}
```

Your task is to:

- In the App, download the list of cities from [this gist](#).
- Be able to filter the results by a given prefix string, following these requirements:
 - Follow the prefix definition specified in the clarifications section below.
 - Optimise for fast searches. Loading time of the app is not so important.
 - Search is case insensitive.
- Display these cities in a scrollable list, in alphabetical order (city first, country after). Hence, "Denver, US" should appear before "Sydney, Australia".
 - The UI should be as responsive as possible while typing in a filter.
 - The list should be updated with every character added/removed to/from the filter.
 - Allow to filter only favourites
- Each city's cell should:
 - Show the city and country code as title.
 - Show the coordinates as subtitles.
 - Show & toggle as favourite
 - When tapped, navigate the map to the coordinates of the city.

- Contain a button that, when tapped, opens an information screen about the selected city.
- Create an information screen for a City
 - Include any data not displayed in the list.
 - May include data for an additional source
 - Organise information as best accommodate user experience.
- Create a dynamic UI that follows the wireframes.
 - Hence, when in portrait different screens should be used for the list and map but when in landscape, a single screen should be used.
- Allow users to select favourite cities.
 - Favourite cities should be remembered between app launches.
- Provide unit tests showing that your search algorithm is displaying the correct results giving different inputs, including invalid inputs.
- Provide UI/unit tests for the screens you have implemented.

Evaluation Criteria

- Provide a README.md explaining your approach to solve the search problem and any other important decision you took or assumptions you made during the implementation.
- You can preprocess the list into any other representation that you consider more efficient for searches and display. Provide information of why that representation is more efficient in the comments of the code.
- Pre-release (e.g. beta) versions of IDEs, SDKs, etc. are forbidden.
- On Android the solution must:
 - Be compatible with the latest Android API.
 - Build Views built with Jetpack Compose.
- On iOS the solution must:
 - Compatible with latest Swift version.
 - Compatible with the latest iOS version.
 - 3rd party libraries are forbidden.
 - Build views using SwiftUI

Clarifications

- We define a prefix string as: a substring that matches the initial characters of the target string. For instance, assume the following entries:
 - Alabama, US
 - Albuquerque, US
 - Anaheim, US
 - Arizona, US
 - Sydney, AU

- If the given prefix is "A", all cities but Sydney should appear. Contrariwise, if the given prefix is "s", the only result should be "Sydney, AU".
- If the given prefix is "Al", "Alabama, US" and "Albuquerque, US" are the only results.
- If the prefix given is "Alb" then the only result is "Albuquerque, US"

Contact us

Feel free to reach out to us if you have any questions or need clarification on any part of this challenge.

- ioschallenge@uala.com.ar
- androidchallenge@uala.com.ar

Submission requirements

- Use a hosted service to share the git repository of your solution (GitHub, Gitlab) and allow public access during the revision.
- We want to see the versioning history as well.
- Please remove public access once we have informed you of the review result.
- Include this document as part of your submission.

Wireframes



