Coding Challenge

Challenge

Your task is to create an **Android mobile app** that uses the Flickr image search API and shows the results in a scrollable view.

This exercise has several functional and non-functional requirements that are listed below.

If you have questions about this task, feel free to send us an email and we will get back to you as soon as possible.

Flickr API

The Flickr API allows you to find all images for a specific search term. Replace **{search term}** in the following URL to get images for the corresponding search term.

```
https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=37ad288835e4c64fc0cb8af3f3a1a65d&format=json&nojsoncallback=1&safe_search=1&text={search term}
```

For example, if you want to get all images related to "*Munich*" you would use the following URL:

```
https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key=37ad288835e4c64fc0cb8af3f3a1a65d&format=json&nojsoncallback=1&safe_search=1&text=Munich
```

Once the request is sent out successfully, you will get a JSON response. Within that JSON response you will find JSON objects that look like this:

```
{
  ”photos”: {
    ”page”: 1,
    ”pages”: 1992,
    ”perpage”: 100,
    ”total”: 199137,
    ”photo”: [
      {
        ”id”: ”54314911413”,
        ”owner”: ”58167807@N00”,
        ”secret”: ”01b52ed72a”,
        ”server”: ”65535”,
        ”farm”: 66,
        ”title”: ”Virtualienmarkt”,
        ”ispublic”: 1,
        ”isfriend”: 0,
        ”isfamily”: 0
      },
      // More photos…
    ]
  },
  ”stat”: ”ok”
}
```

To get the actual image data, you must use the farm, server, and id values from the JSON object above and put them into the following URL:

```
https://farm{farm}.static.flickr.com/{server}/{id}_{secret}.jpg
```

For the given example, the image URL looks like this:

```
https://farm66.static.flickr.com/65535/54314911413_01b52ed72a.jpg
```

If you need more information about the Flickr API, please visit the Flickr API documentation.

If the provided API key (`37ad288835e4c64fc0cb8af3f3a1a65d`) is invalid for some reason, please feel free to get one by yourself here.

Acceptance Criteria

Functional Requirements

- The searched results should be shown in a scrollable view.
- The user can see the search history.

Nice to have:

- Consider how to improve the general user experience with your expertise.

Non-Functional Requirements

- The most important functionalities are unit tested.
- Errors and edge-cases are considered.
- Best practices regarding architecture and code style are considered.
- You are allowed to use any 3rd party libraries.
- Use Jetpack Compose for building the UI.

Deliverables

We do not expect a perfect solution. Our goal here is to assess how you approach technical assignments such as this one.

Implement your solution as an Android Project using the Android SDK. Please send the following items via email:

- The source code of the project (as a ZIP file or a GitHub link)
- The compiled APK