Candyspace iOS tech test

Please put together a small iOS application in Swift that shows a gallery of images from a search input using the https://pixabay.com API.

API

The spec and responses are documented at https://pixabay.com/api/docs/

You can use the following API key for all your requests

API Key: `13197033-03eec42c293d2323112b4cca6`

An example request might look like:

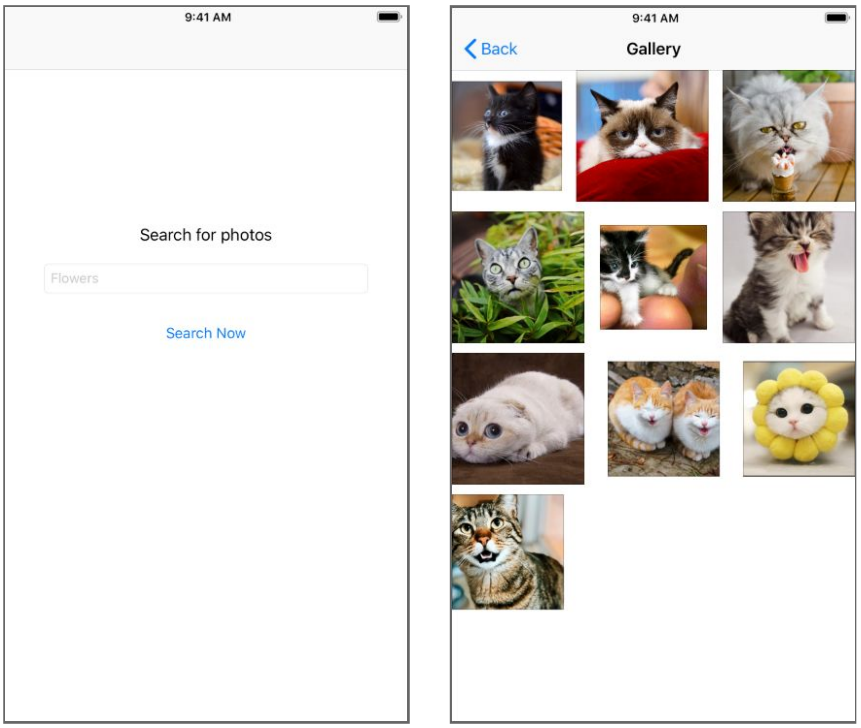
https://pixabay.com/api/?key=13197033-03eec42c293d2323112b4cca6&q=yellow+flowers&image_type=photo

UI

Your application should contain two UIViewControllers. One that contains a text input for the user to enter some kind of search criteria along with a button to initiate the search.

When the search button is pressed it should take you to the gallery page which under the hood could be using UICollectionView or a UITableVIew it's entirely up to you, however a UINavigationController should be used to navigate between the two screens.

The designs are completely flexible and could look however you want them to look like, feel free to release that inner designer! An example of a very basic design could look like the following:



You should cache the results for a particular search term in memory so that the results are available instantly if the same search is performed again.

You should cache the downloaded images in memory so that they do not need to be re-requested.

If the application receives a memory warning you must handle purging objects from the cache.

You can architect your solution however you like. Feel free to expand on the above requirements and ad some of your own additional features or flourishes.

Unit Tests

This section is optional, but as good developers we should strive to have testable code. Hopefully you would have architected your solution in a manner that would allow it to be unit tested so adding some tests shouldn't be too much more work. The caching layer would be the most appropriate to start with.

Constraints:

- The application only needs to support iPhone in portrait orientation.
- The design should scale to work with all iPhone screen sizes.
- You should NOT use any third party dependencies.

- You don’t need to implement specific error handling for all failure cases when calling the API, a generic UIAlertViewController error is fine. (Just don’t crash!)

Deliverable

Your solution should be delivered as a zip file containing an Xcode project with complete source code.