**Readme**

**Installation:**

- The app uses SDWebImage as a dependency. Install pod ‘SDWebImage’ in order to run.

- App runs with no known bugs or compiler warnings. The API key is in Networking/NetworkConfiguration.swift. Sometimes no photos are returned because requests exceeded API limit.

Some decision explanation. Not needed to run app.

**Feedback:**

The code is pretty straight forward there are different ways to implement the viewModel, I used a property observer and closures to achieve data binding.

- Pagination: This was a naïve approach, this was done using the page number in the API, the problem I encountered is that you must fetch the max photos per page(30) or the next page will contain photos from the last page. Even though the code checks for duplicates, some pages will contain all duplicates and it’s a bad user experience.

- Tests didn’t cover too many cases like edge cases. Tests simply ensured that the code should work as intended.

- For the photo grid list, there’s a method in the Extension.swift file that implements image resizing. I did not use it because the images looked betted when aspect fill is used.

- The notification communication pattern was used for simplicity because the app had few features. However ideally a delegation/ observer pattern would be better, because it’s hard to keep track of which controller is firing and listening.