Melissa Zellhuber

Features

A reachability class which confirms the users' internet status is implemented, ensuring the user won't assume the app is not functional if their device is not connected to the internet via cellular data signal or wi-fi.

3D Touch has been incorporated in order to permit the user a peek of the full image. By swiping up from the peek, a share button is provided. Use of an `ActivityViewController` allows the operator to either save the full photograph to the device library, or to share through any accounts that are linked to the device. Saving the photo will prompt the user to confirm permissions, which are required for this action.

Infinite scroll provides the user with a familiar and smooth experience. 50 photos are loaded with the initial call, and upon the user scrolling to the end of the screen, a further 50 will load. This feature makes use of `UICollectionViewDataSourcePrefetching`. This is coupled with a cache, which ensures the photos can be scrolled through without visual anomalies caused by asynchronous loading.

A detail view provides metadata about the photo, such as the title and a link to the original Flickr of the author. The detail view also provides scrollable access to other photos shared by the user. Additionally, if the image author's name is clicked, their profile is shown and can be explored.

The user is provided with a search bar, providing quick and reliable access to the content they want. The results returned can be ordered by time or relevance, which is accessible by use of a standard segmented control.

Each type of image display provides the user with the photo in its original format. This is of particularly relevance in a photo sharing app. The quick preview screen displays the entire photo in as much detail as possible whilst maintaining aspect ratio.

A message with an emoji will appear if no results were available. This utilizes `TBEmptyDataSet`. Loading visualization and notification messages are handled by `MBProgressHUD`. These provide further visibility of system status.

The app's initial search for food is available via a button to the right of the search bar, simplified pictorially by a kitchenware metaphor. During initial loading, a welcoming splash screen is displayed.

Future Improvements

Given more time, there are multiple improvements I would make to the app. One straightforward improvement would be to enable preview and sharing on the detail screen. Further, I would provide the facility to allow the user to save their favourites. This would require an account system, which would include authentication. Another distinguishing improvement would be to integrate multiple photo sharing websites, offering the user more than any single source could. This could motivate the user to create an account and hence increase the app's desirability.

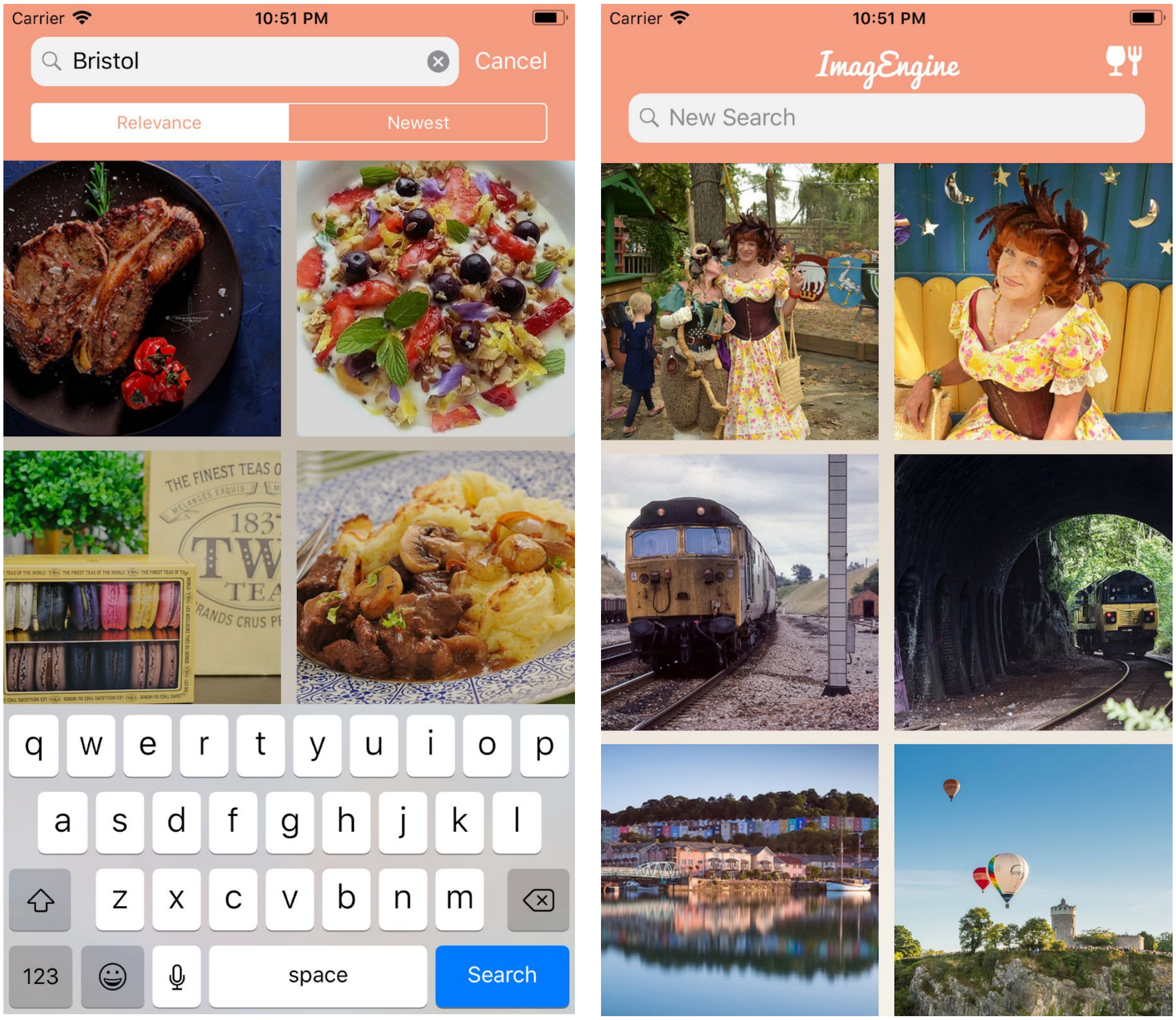
The consumer may find a search history feature useful. This way, if they neglect the favourites feature, the consumer can still get back to a photo they wish to share without hassle. Photo editing or captioning would be made possible on photos prior to sharing them, increasing the usefulness of the app further. The user may wish to interact with the original source of the photo, and as such upload their own photos to Flickr (or, if extended, other photo sharing websites integrated with the app). The food button could be expanded to provide a category-based browsing system, with the categories decided upon by research upon the expected demographic, such as a focus group and survey.

Structure

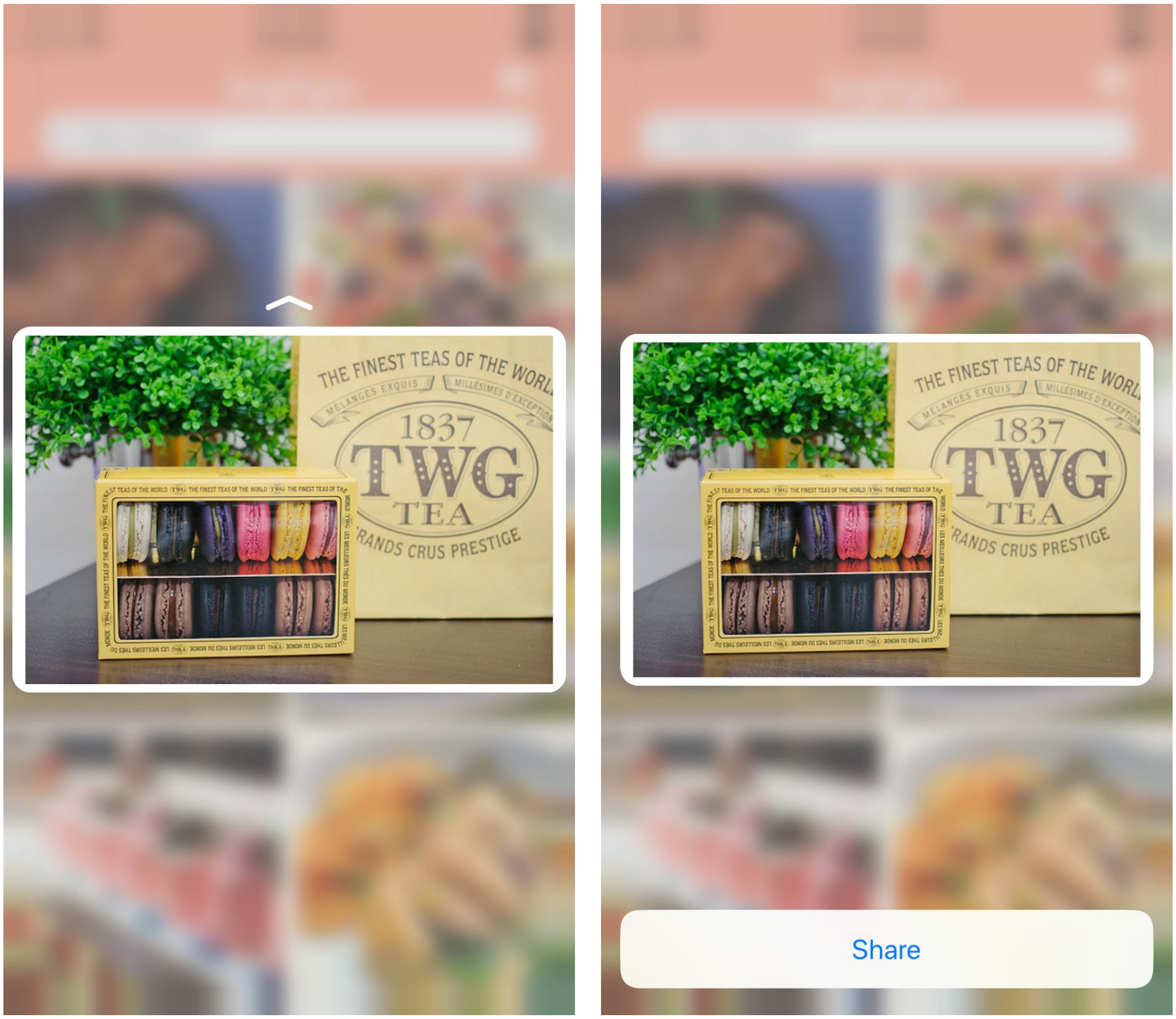
The project's version management is handled by Git. CocoaPods has been employed for dependency management.

A small networking class has been written in order to avoid the huge data overhead created by the requirement for Alamofire or another such networking library. There will only be infrequent API calls, so there were significant disadvantages at this stage of the app's functionality.

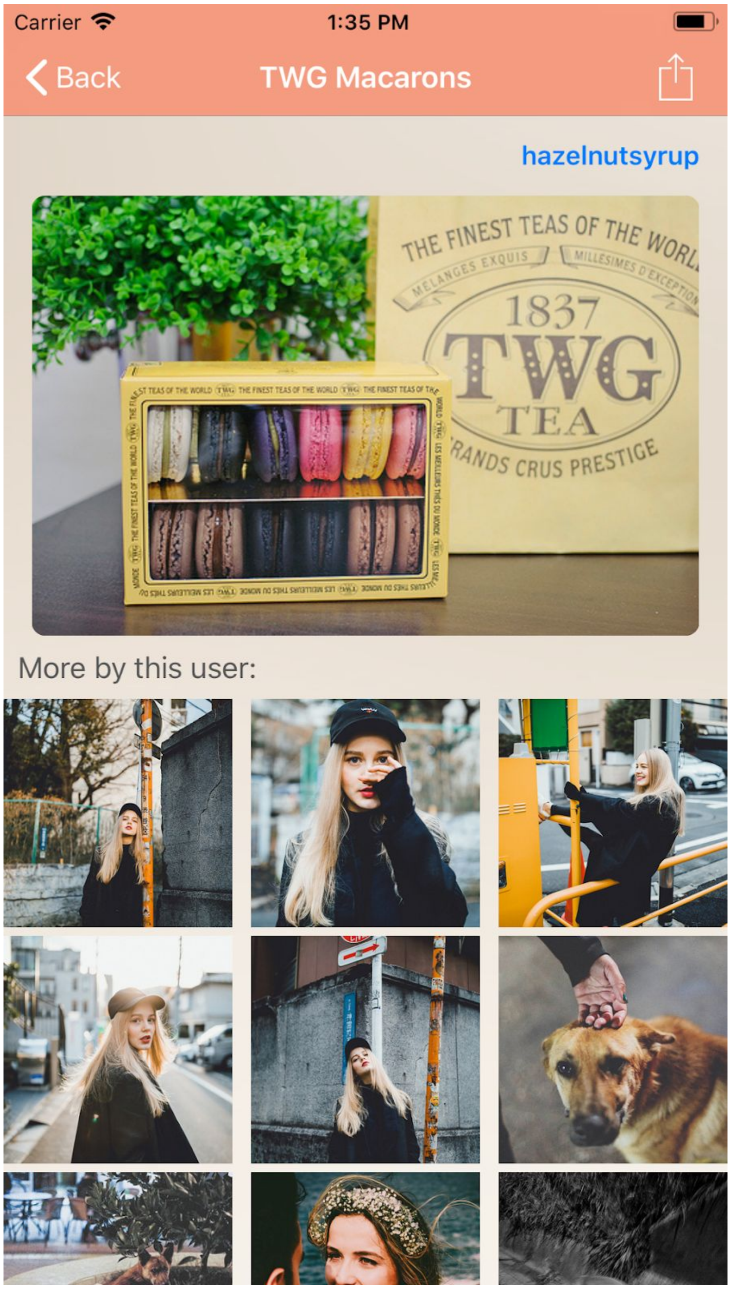
A `Codable` protocol is used to receive data from the API and transform it into model structures. Using this methodology facilitates easy access of the properties of the incoming data. `JSON` for each item is received from the network call, and as per the protocol converts to a model structure.



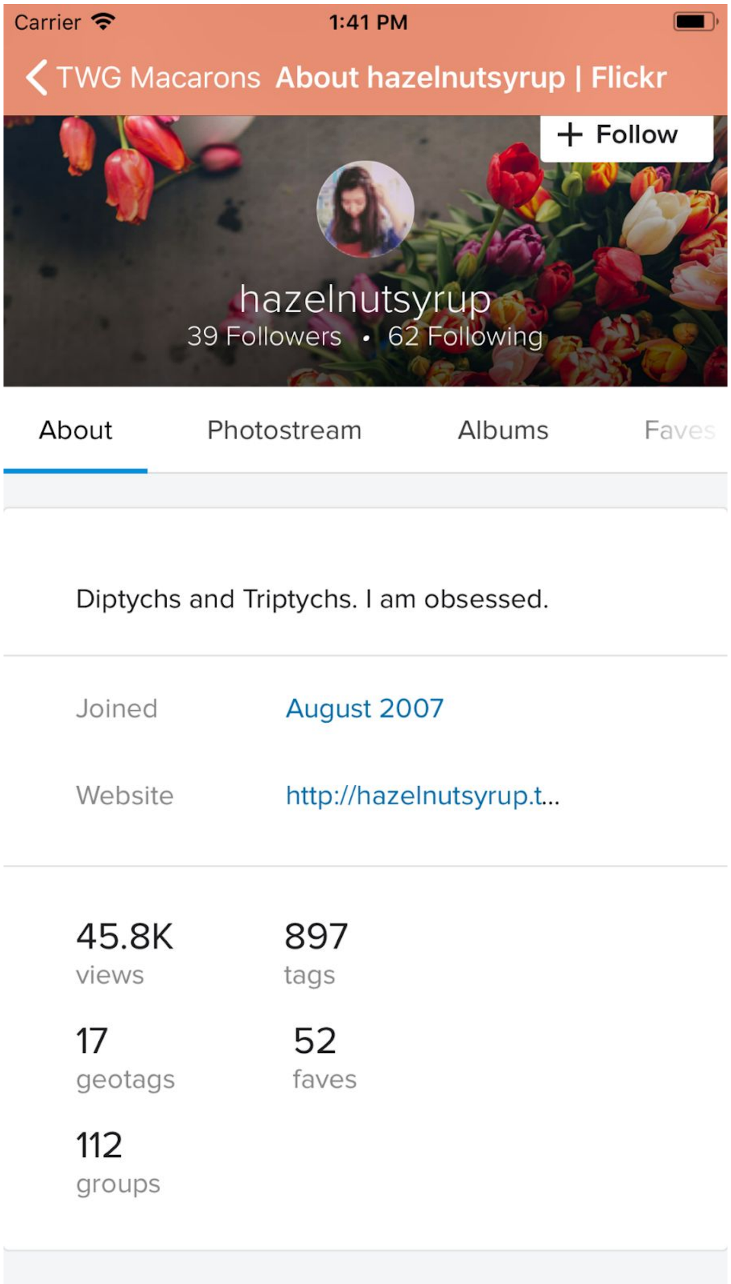
An example search. Note that food search is available via the kitchenware metaphor



Quick view draws focus to the image. Share functionality is provided by swiping up from quick view



Detail view provides easy access to the author’s other photos. Scrolling is used to navigate this screen. Profile view can be activated by clicking the author’s username



hazelnutsyrup

39 Followers • 62 Following

Diptychs and Triptychs. I am obsessed.

Joined August 2007

Website http://hazelnutsyrup.t...

45.8K views

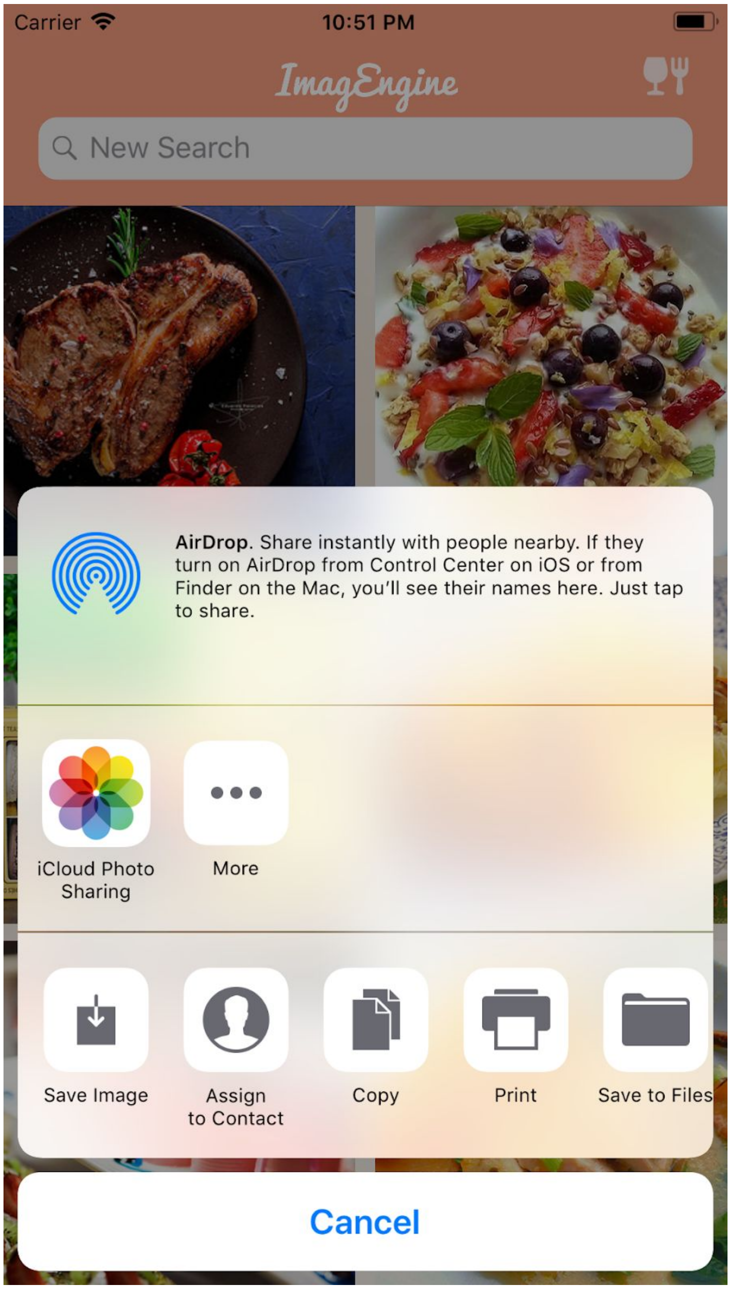
897 tags

17 geotags

52 faves

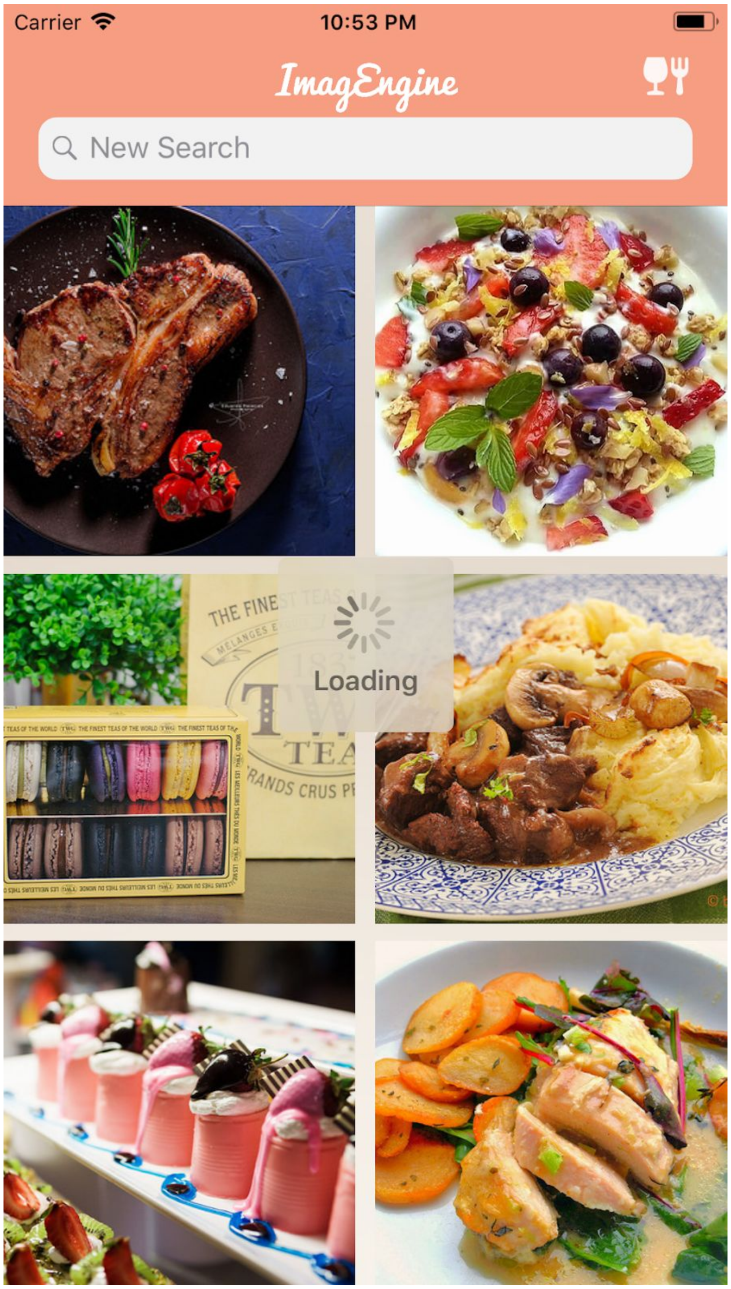
112 groups

Profile view. All features of the profile can be explored

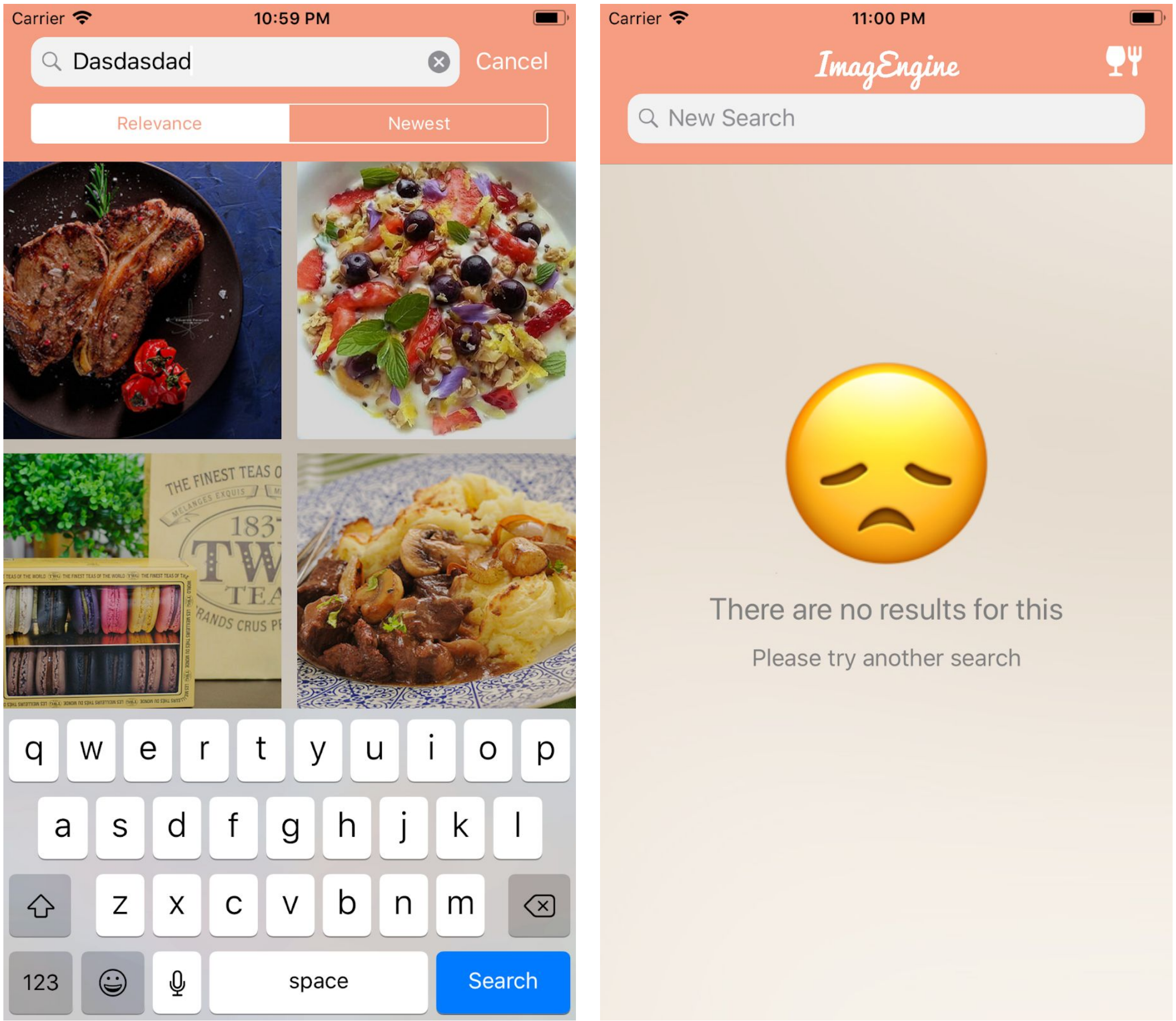


AirDrop. Share instantly with people nearby. If they turn on AirDrop from Control Center on iOS or from Finder on the Mac, you'll see their names here. Just tap to share.

The user can save or share photos they have found. If there are social media accounts linked to the device, access to them will be displayed the menu shown above



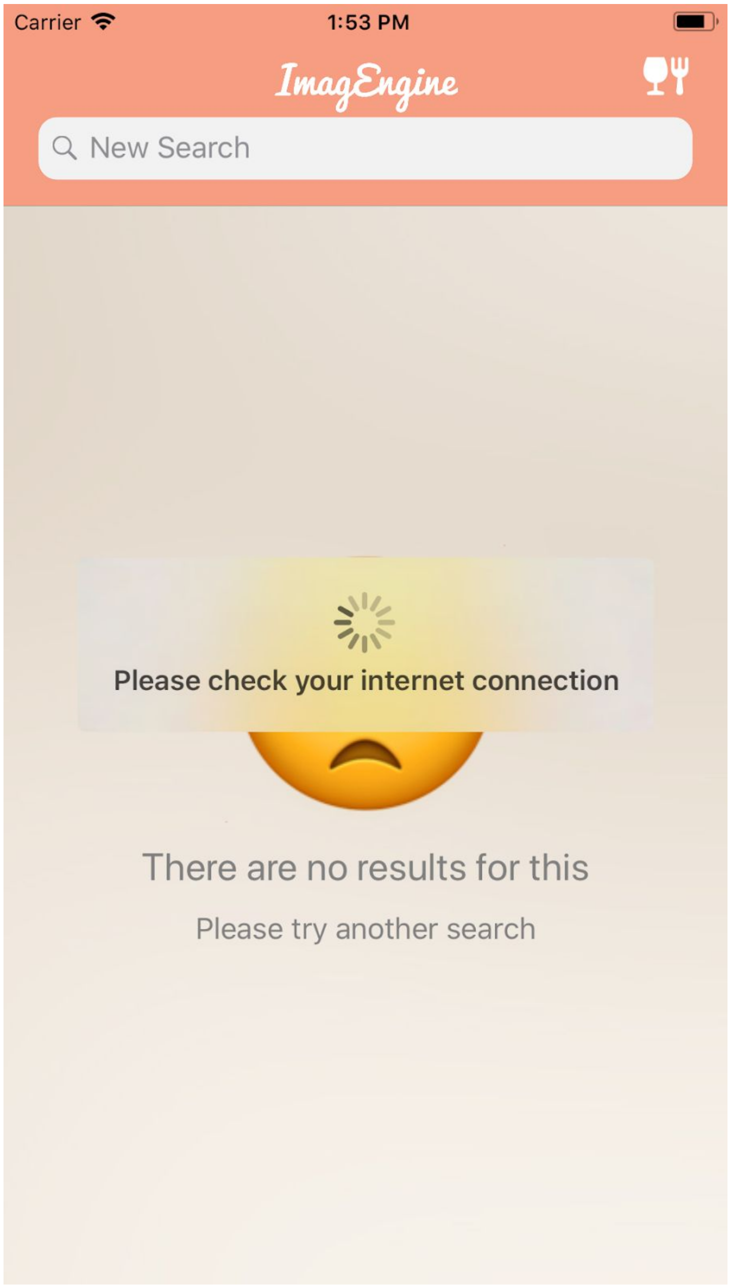
Loading visualizations provide visibility of system status



There are no results for this

Please try another search

Responsive no-result searches are handled



Please check your internet connection

There are no results for this

Please try another search

Reachability status confirmation avoids confusion over app status