PICSUMPREVIEW

1. Introduction

The purpose of this document is to capture the high-level design and use cases of the iOS mobile app named "**PicsumPreview**" developed in Swift 5 with XCode 11.4 to run in iPhone and iPad for Portrait as well as Landscape mode. The minimum deployment target for this app is iOS 11.4.

2. Requirements Detail

Please find the below task, you should more focus on design, memory management, and Unit test cases.
Note: Please do reach us for any queries.

Kindly find the below task and revert back once you completed. Develop an application using Image Feed https://picsum.photos/
Picsum api doesn't need any authentication. So it should be easy to consume this api.
Use https://picsum.photos/list to get the list of images.
Use https://picsum.photos/200/300?image=<imageId> to get a specific image
Choose any Good UI approach that is up to your imagination and creativity to show all images
App should load images as user scroll
App should support Portrait and landscape orientations
Use some cache mechanism to load images
Must Requirements
Swift should be used as a language to develop this application
Clean
Setup dependency management for any third party frameworks required for this application
Additional Bonus points
Write Unit tests

3. Project dependencies

COCOAPODS used as dependency manager and the following dependencies used to build the mobile app:

SwiftLInt – Used as linting tool to capture warnings in XCode. The 'swiftlint.yml' file used to configure for this project. Line length set as 160 and Pods lint warnings ignored.

Kingfisher – Used for image downloading/Caching from given URL, Image extension add to set image from URL. ImageCacher singleton class created using it cache limit configured as 300 MB for memory cache and 500 MB for disk cache.

Lotti-ios – Used to show nice circle animation while loading, 'loading-polichat.json' use as animation source. LoadingAlertView created to show and hide animations.

4. Design patterns

This app uses MVVM as the structural pattern and has two folders **PicsumPreview** for development and **PicsumPreviewTests for** testing:

4.1 PicsumPreview

PicsumPreview folder has the development code in the following folders:

Model:

PicsumImageModel to store transformed API response using coding keys

View:

Collectioview's ColumnFlowLayout – Caculate dynamic size of column
Collectioview's Datasource – Provides data to display on Header and Cell, sections are ordered in ascending order of Author's images.
PicsumImageViewCell – has the UImageview to display photos
SectionHeaderView- Displays the Author name as sticky header
LoadingAlertView – Creates and handled loading animation
AlertView – Display alert on error conditions like API, Network, Data failures
UIImageview extention – Uses Kingfisher to set image from URL

ViewModel:

PicsumImageViewModel which holds array and authors array. It provides methods to get the author count, author name for the section header and Image id for each row.

Controller:

PicsumImageCollectionViewController – Manages the interaction between, user action, system actions, view model and API service.

Networking:

PicsumAPIClient – implements imageList data task method to retrieve the image list from https://picsum.photos/list end point.
ImageCacher – It uses the default cache provided by Kingfisher framework for caching and retrieval images using URL absolute path as cache key.

Launch:

AppDelegate – default implementation.

Utils:

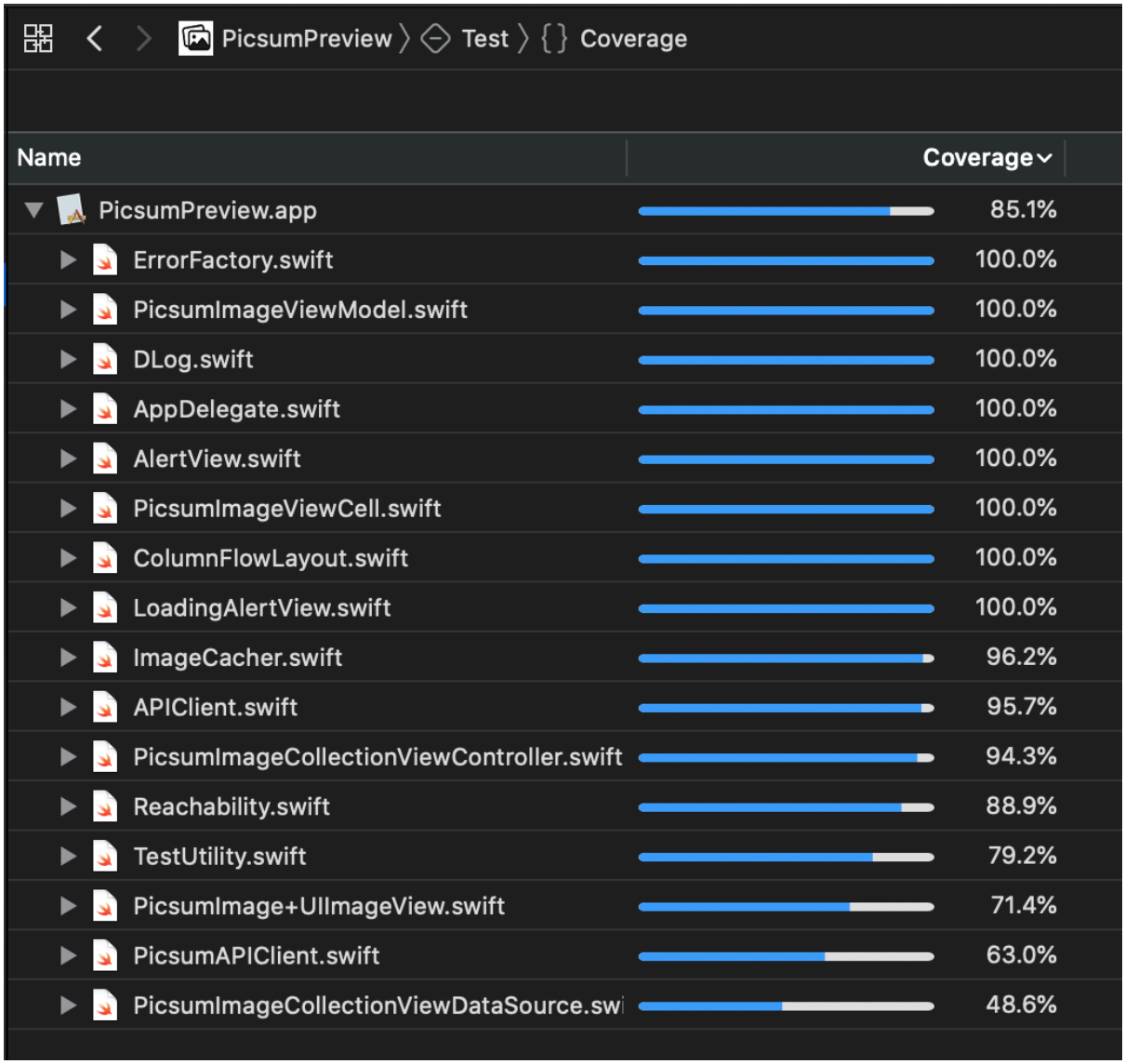
ErrorFactory – enum which Creates error object based on error code
DLog – prints debug mode log to the console
Reachability – checks reachability of https://picsum.photos via phone network.

Resources - Contains:

Storyboards for launch and main screen, image assets, info plist, constants, animation .json file, and localization strings.

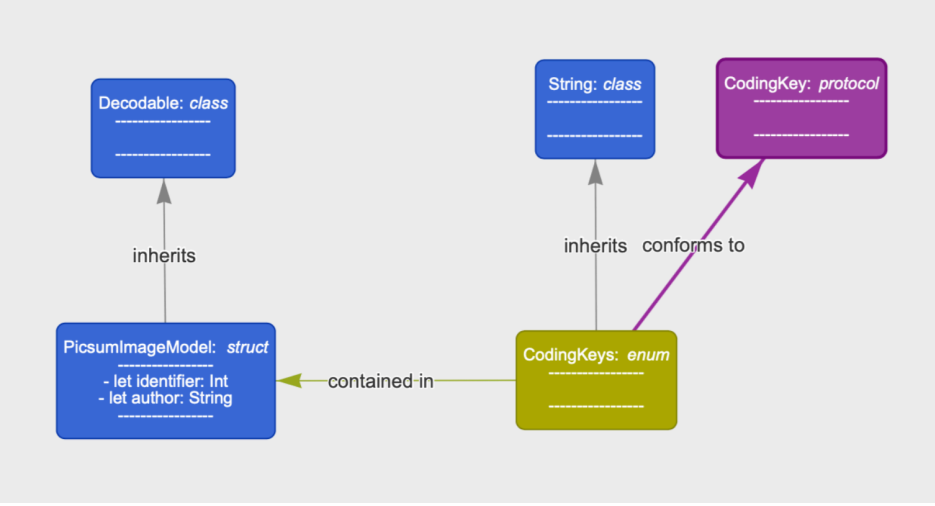
4.2 PicsumPreviewTests

PicsumPreviewTests folder has the unit testing code, unit test coverage is 85% and below image has the percentage covered for each file.

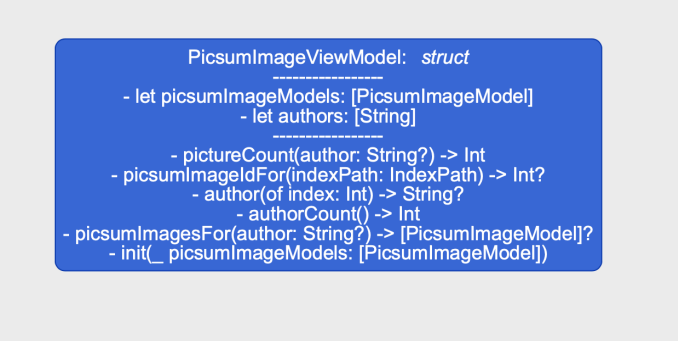


5. Class Diagrams

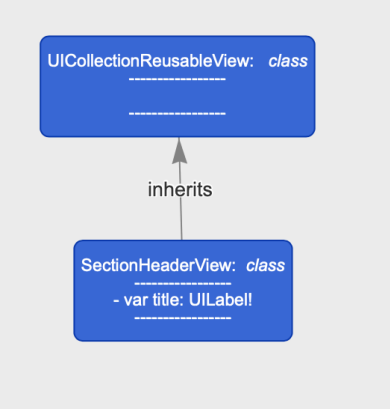
5.1 PicsumImageModel

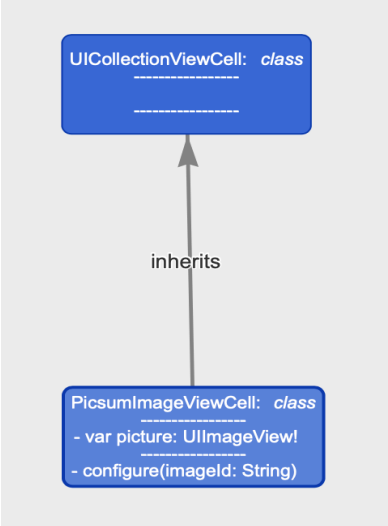


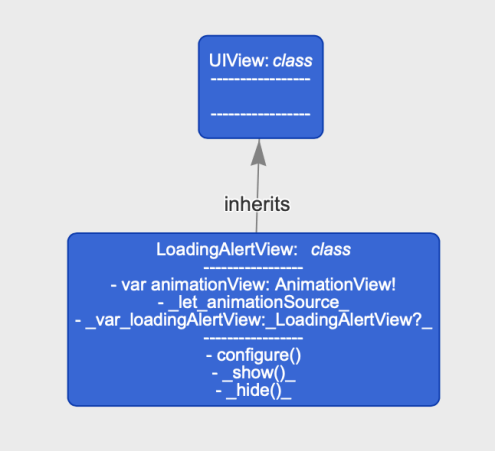
5.2 PicsumImageModel



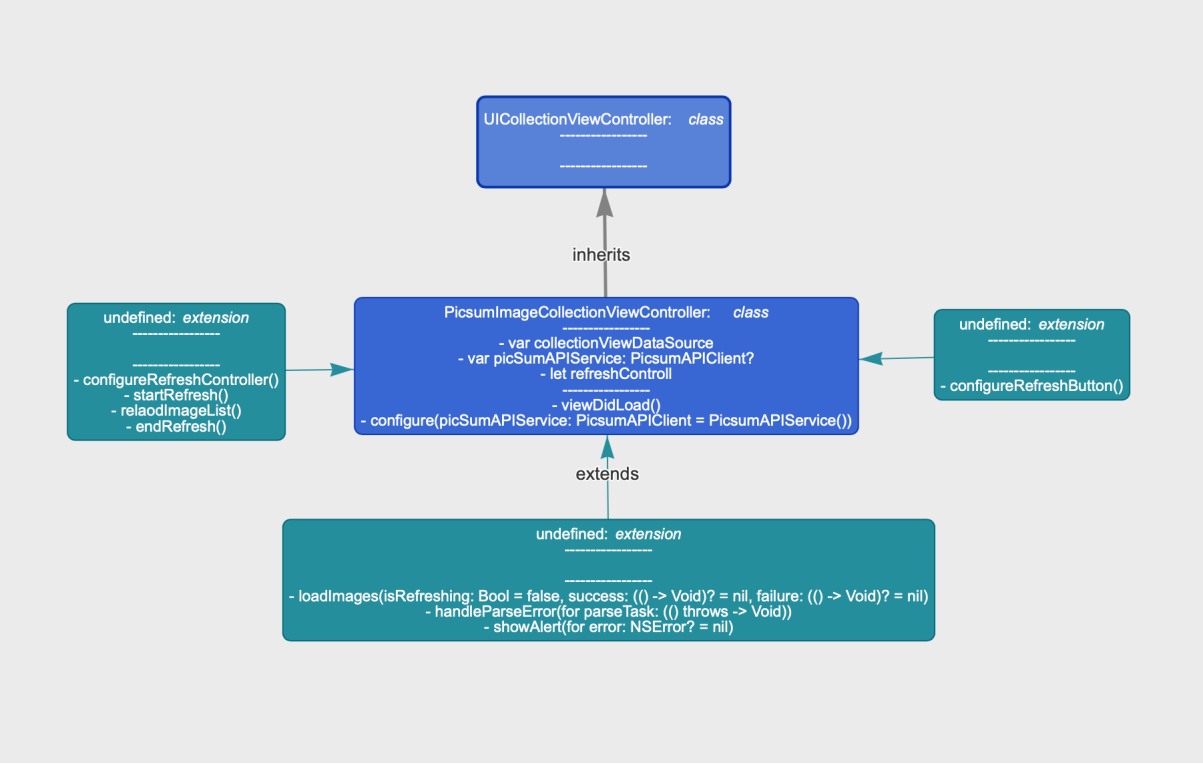
5.3 View



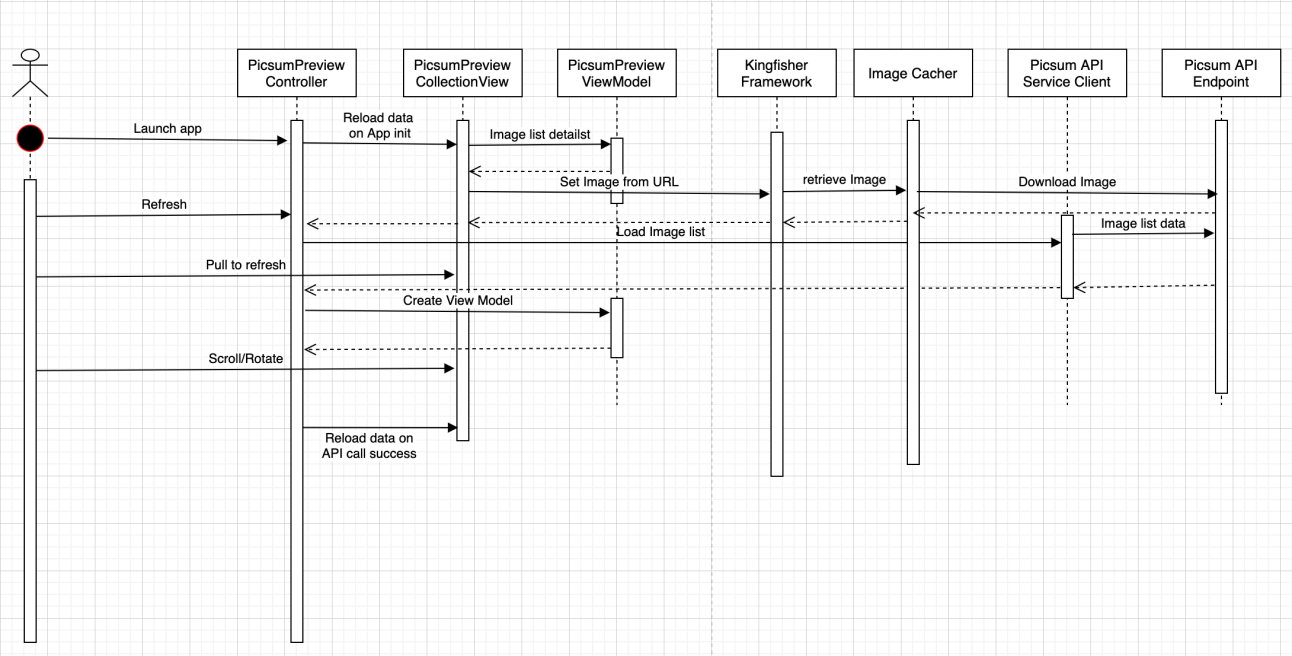




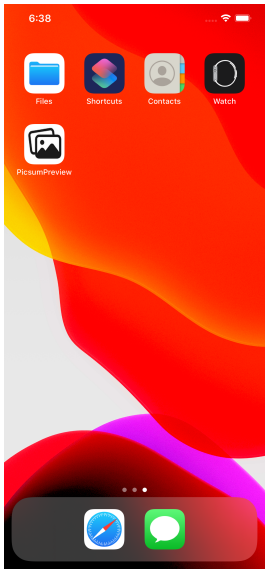
5.4 View Controller



6. Use case sequence diagram

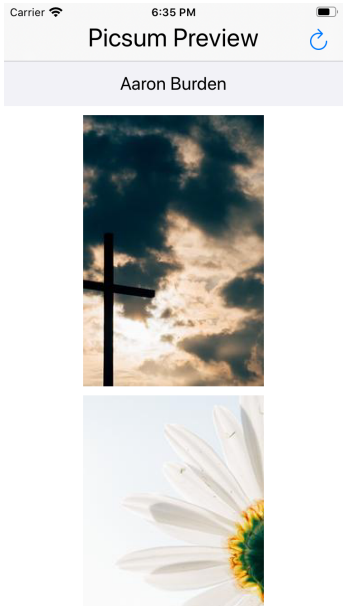


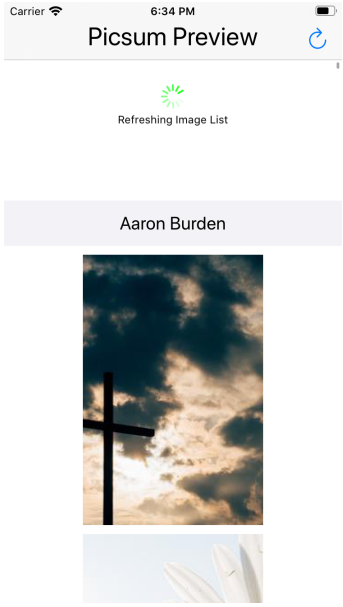
7. App screens

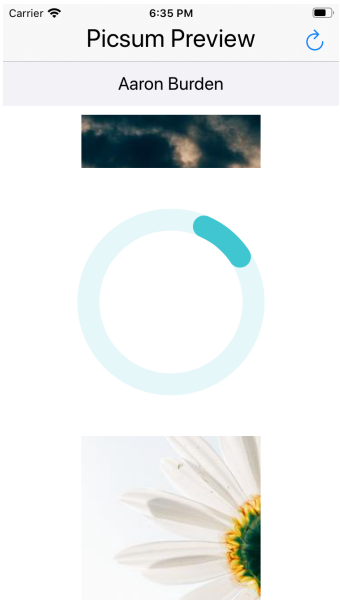

6:38
Files
Shortcuts
Contacts
Watch
PicsumPreview


6:37
Lorem Picsum
The Lorem Ipsum for photos.


6:37
Picsum Preview


Carrier
6:35 PM
Picsum Preview
Aaron Burden


Carrier
6:34 PM
Picsum Preview
Refreshing Image List
Aaron Burden


Carrier
6:35 PM
Picsum Preview
Aaron Burden

Picsum Preview

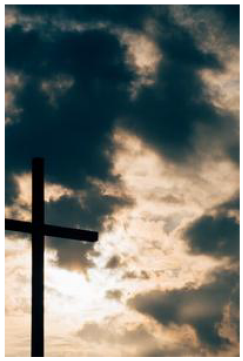
Aaron Burden

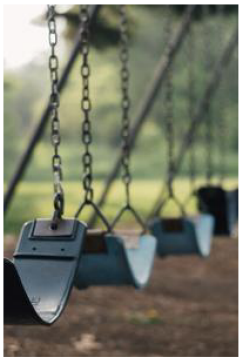
6:46 PM Fri Jul 3

100%

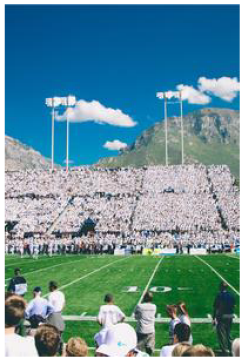
Picsum Preview

Aaron Burden



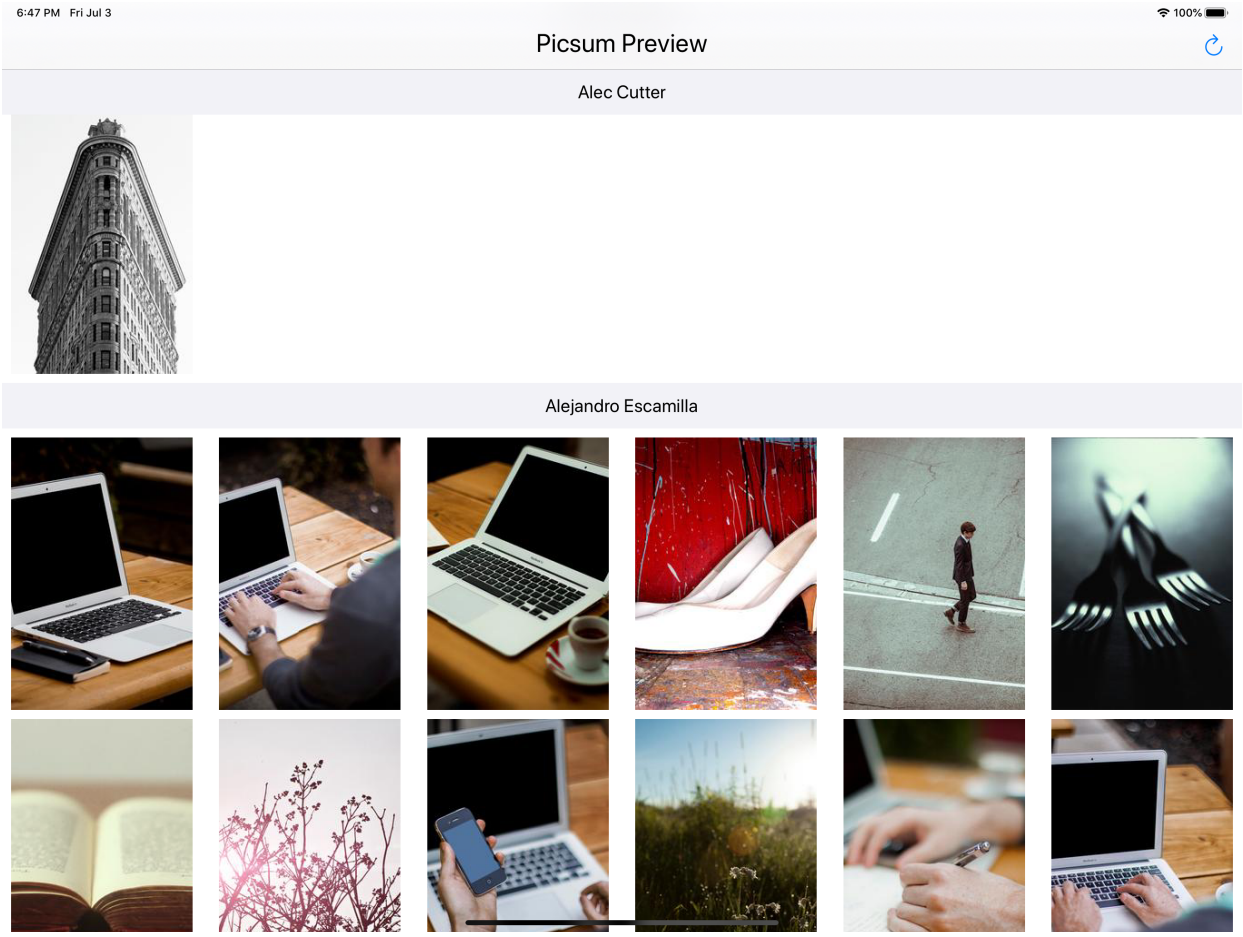


Abigail Keenan




Adam Przewoski

Adam Willoughby-Knox


6:47 PM Fri Jul 3
100%
Picsum Preview
Alec Cutter
Alejandro Escamilla

Thank you!