Wallpaper Collection Documentation

Filipa Ivanković & Lucijan Pavić

6th December 2022

Product Definition Statement

“Wallpaper Collection” is an application for downloading images to be used as wallpapers on iPhones & IPads. It shows both locally stored images and images retrieved from the Unsplash API. Users can view a collection of these images and download them (after giving the necessary permission for that). Users can also favorite these images for an additional method of viewing them. For some ease of access, the user can also search through the wallpaper grid list to find desired images. The application is suited for the IPhone 13/12/11 and IPad. Older versions of the IPhone may still work, but due to the View layouts, they may not be best suited for it.

List of features

* Downloading images
* Locally stored images
* Images provided by the Unsplash API
* Favorite page
* Add images to favorites by simply clicking on a button
* Deleting all Favorited images
* Deleting individual Favorited images by clicking on the same button used to favorite the image
* Refreshable
* Search and filter wallpapers by name or tag
* Lottie animations
* Favoriting,
* Unfavoriting
* Downloading
* Getting images from Unsplash API

Documentation

WallpaperCollectionApp

* Creates the “ContentView” and sets the Environment object for the “wallpaperLocals” which contains all of the locally stored wallpapers

View

ContentView

* Creates the tabs for the nav menu and initiates the project

TabView - CustomTabView.swift

WallpapersView

Displays all of the images loaded from the Assets folder and from the Unsplash API. The images are displayed in a grid. The user can also filter the images using the search feature located at the top of the grid.

WallpaperDetailView

This screen is accessed by pressing/tapping on a wallpaper stored locally within the grid display. Displays the full screen images from the Assets folder. Each image comes with a “Favorite/Unfavorite”, “Download” and “Additional info” buttons. Pressing the favorites/unfavorites button will add/remove the image from the “FavoritesView” page. “Download” button will download the wallpaper to the IPhone/IPad “Photos”, if the user allows the app to store images there. The “Additional info” button will showcase the images given name and a few tags associated with that wallpaper.

WallpaperApiDetail

Displays the full screen images from the Api with buttons for favorite page and download. The API alternative page for the “WallpaperDetailView”, as this file only interacts with the API images due to the API images having a very different structure from the one’s stored locally. The “Favorite/Unfavorite” and “Download” button work just the same way as in the “WallpaperDetailView”. The “Additional Info” button is not present as the Unsplash API does not store name, tags, or anything about the image itself into it; therefore, making the “Additional Info” button unnecessary.

FavoriteView

Displays a page of favorite images from the Api as well as from the Assets folder. It also allows the user to remove all images from the favorite array. The user can erase favorite images individually by going to each image's “WallpaperDetailView”/”WallpaperApiDetail” views, but here, there is a button to remove all from the “Favorite” page. The user will be prompted with a confirmation dialog window to remind them that they are going to be deleting data. This confirmation was added in the case that the user may accidentally click on this button.

LottieView

Used to create Lottie animations for the favorite/unfavorite button and download button. These animations are featured in the “WallpaperDetailView” and “WallpaperApiDetail”.

Styling

Additional folder for styling structs to reduce code repetition and avoid DRY.

CircleButtonStyle

Creates the styling for the buttons in the detailed views.

Model

ImageSaver

Used to save images to the IPhone/IPad gallery for the “Download” buttons.

WallpaperApi

Gets a json form the Unsplash API and decodes it so that the data can be used for displaying the wallpapers. The key and URL necessary for the API are stored here as well. The API gets a defined number of random images from the API for displaying and downloading purposes. Due to the API’s restrictions, we are only able to display and fetch a maximum of 30 random images per request. We are also limited to 50 API calls per hour, due to, again, the API restrictions.

WallpaperLocal

Creates a model class for the data in the data plist. It creates the data mutators and accessors for the wallpaper images stored in the “data.plist”

WallpapersLocals

Reads data from data plist and stores them in an array that is accessed by the “WallpaperDetailView”.

Third-party frameworks

LottieFiles

Open-source framework used for displaying the animations used for the Wallpaper Collection app. The animations are contained in the “Animations” folder in the form of JSON files. We used animations for the “favorite”, “unfavorite” and “download” features of the app. We worked with the 3.5.0 version of Lottie files.

Links:

<https://lottiefiles.com/integrations>

<https://github.com/airbnb/lottie-ios.git>

SDWebImage

“SDWebImageSwiftUI is a SwiftUI image loading framework, which is based on [SDWebImage](https://github.com/SDWebImage/SDWebImage).

It brings all your favorite features from SDWebImage, like async image loading, memory/disk caching, animated image playback and performances.

The framework provides the different View structs, which API match the SwiftUI framework guideline. If you're familiar with Image, you'll find it easy to use WebImage and AnimatedImage.”

This framework is used to display the API Unsplash images in the Wallpaper and Favorites grids

LINK:

<https://github.com/SDWebImage/SDWebImageSwiftUI.git>

Unsplash API

The API that was used for displaying the online images as wallpapers. The images are randomly selected from the API, so each time the application is started, they may be a different set of images from the API displayed. Because we are free users of the API, we had to, and have to work with a few restrictions on how many requests and how many images we can fetch. The maximum number of images we can get with one call is 30. The maximum number of times we can call the API within an hour is 50. In the case that the API images are not shown, it is very likely that the number of requests has reached a maximum amount. If that happens, changing the key, or just waiting an hour should resolve the problem.

The amount of images can be set in the “WallpaperAPI” file in this line of code:

**let** url = "<https://api.unsplash.com/photos/random/?count=3&client_id=\(key)>"

Just change the “...count=NUMBER&...” with a different number up to 30.

Valid keys in the case the API requests get “used up”.

Key 1:

1MZdofELglOwog1WWe8\_c48GZYuStRSogWq9cK1JqEA

LINK:

<https://unsplash.com/developers>