# README

## Overview

This project is part of the OpenClassrooms Learning Program and available on GitHub

The main purpose of this project is to create the first application of the learning program.

### Features

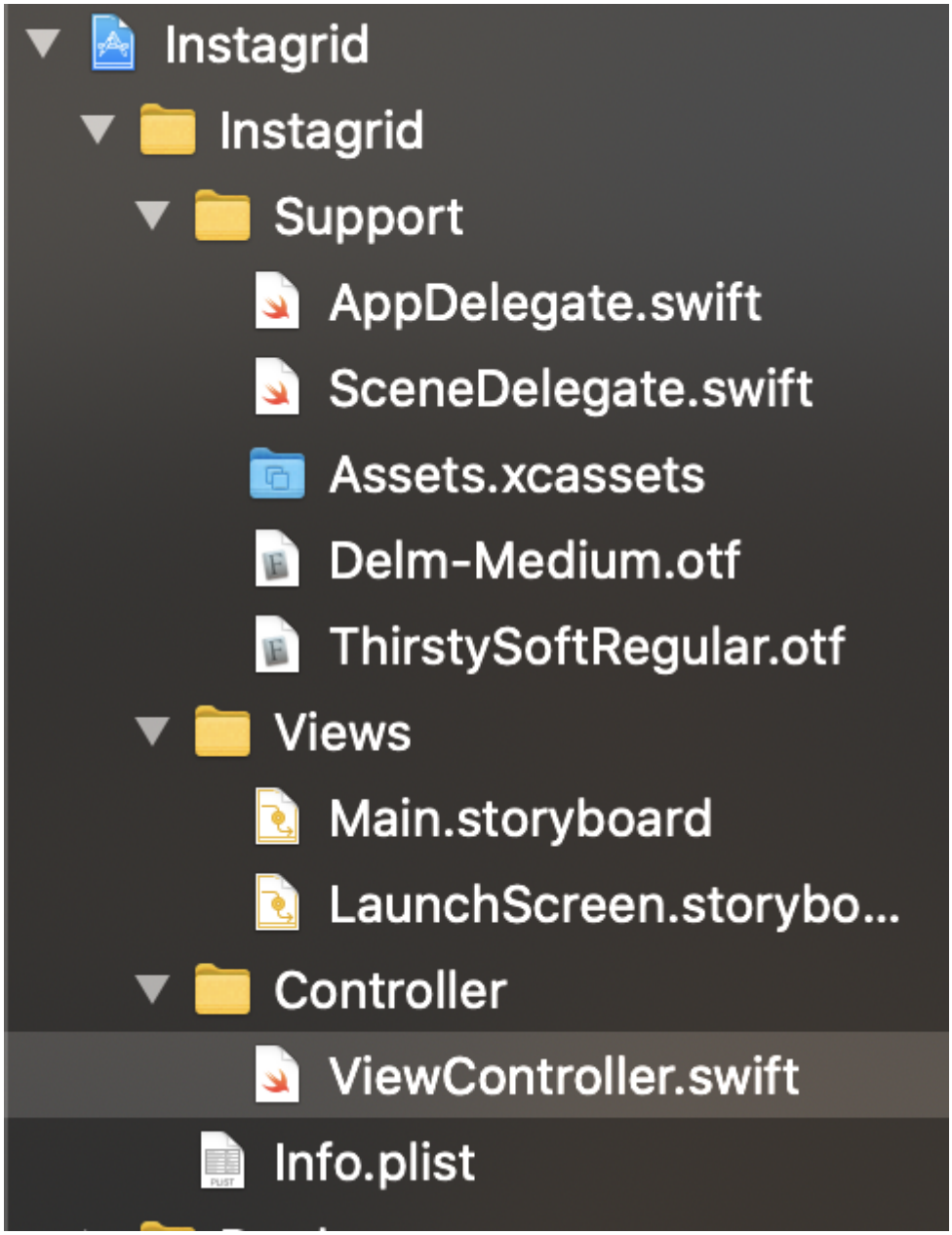
The application allow the user to combine 3 or 4 photos from their library. They are able to choose between 3 different layouts. The render is a square image that can be shared with their friends.

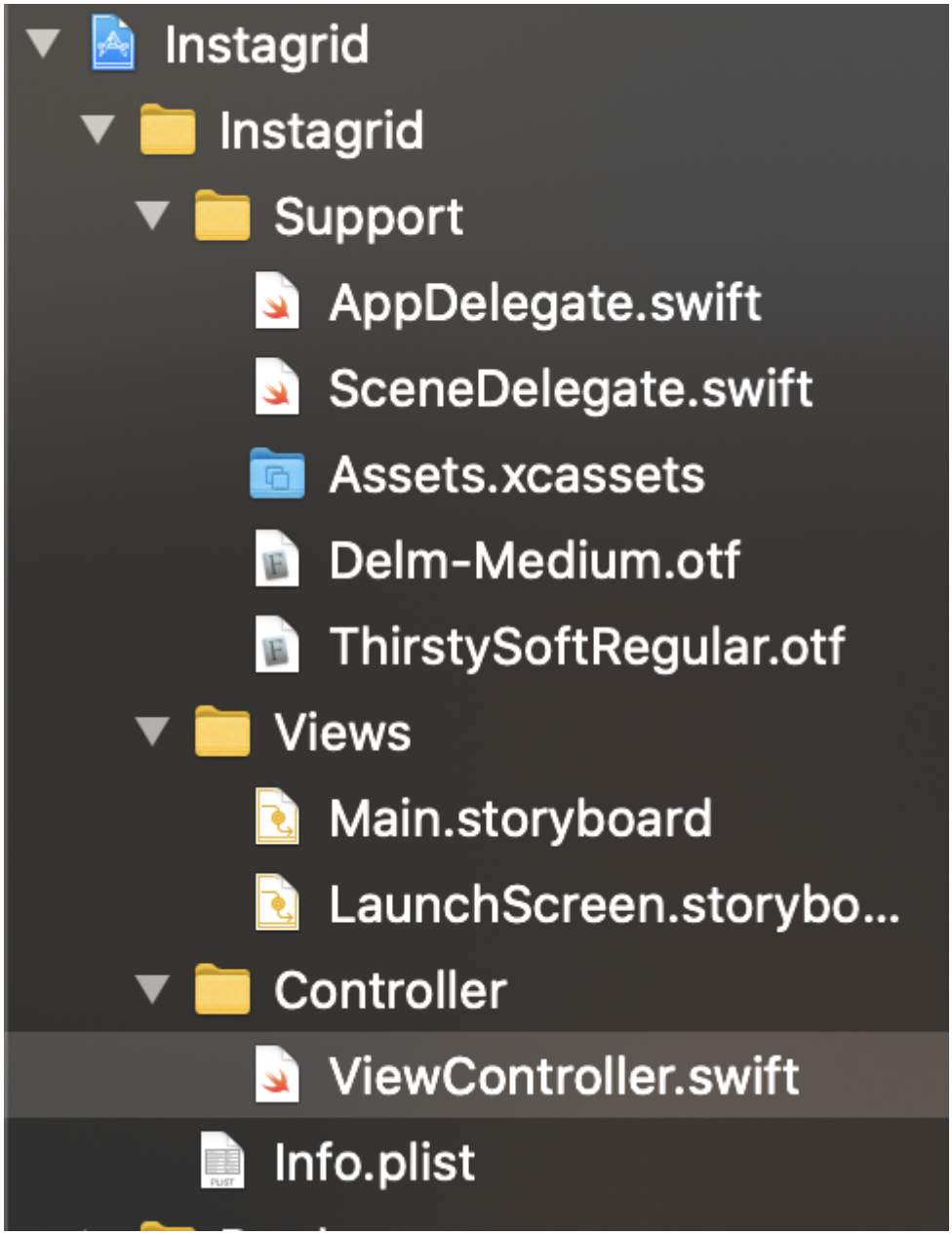
## Architecture

The architecture used in this project is the MVC architecture.

There is no Model needed here. Only the View and the Controller exists.

The final project is classified in three folders :


Instagrid
Instagrid
Support
AppDelegate.swift
SceneDelegate.swift
Assets.xcassets
Delm-Medium.otf
ThirstySoftRegular.otf
Views
Main.storyboard
LaunchScreen.storybo...
Controller
ViewController.swift
Info.plist




- Support : The files contained here are the files created automatically by Xcode, such as AppDelegate.swift, SceneDelegate.swift and Assets.xcassets.
  Two font files were added to fit the project necessities.
- View : All the files related to the views of the project. The design was created in the storyboard, and a LaunchScreen has been set too.
- Controller: There is only one file in this folder : ViewController.swift. More information are detailed

below.

## View Controller


Outlets
Properties
Actions
Lifecycle Methods
Photo Logic
Swipe Logic
Layouts
UIImage Picker Controller

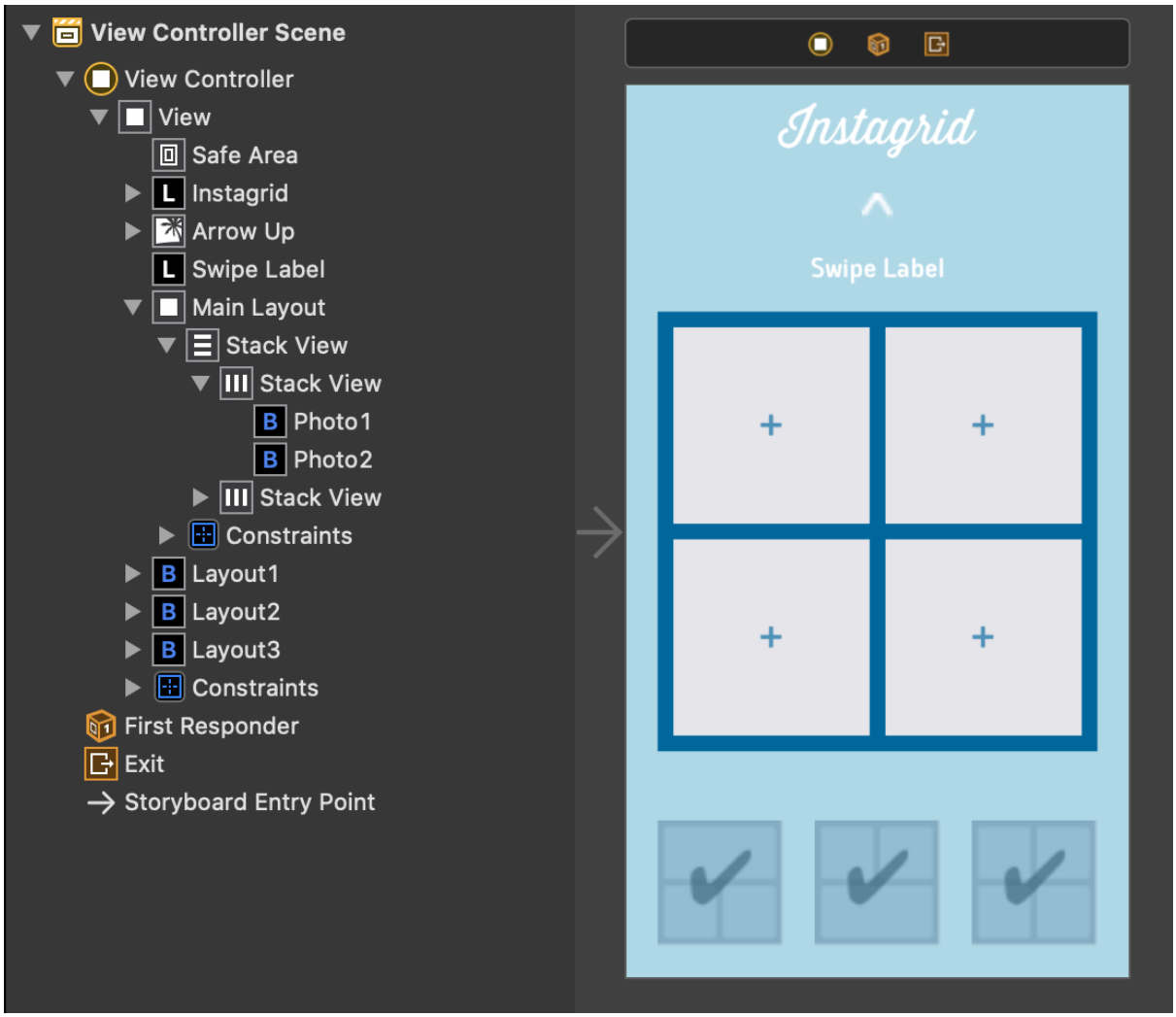
The different parts of the ViewController is detailed by the Marks.

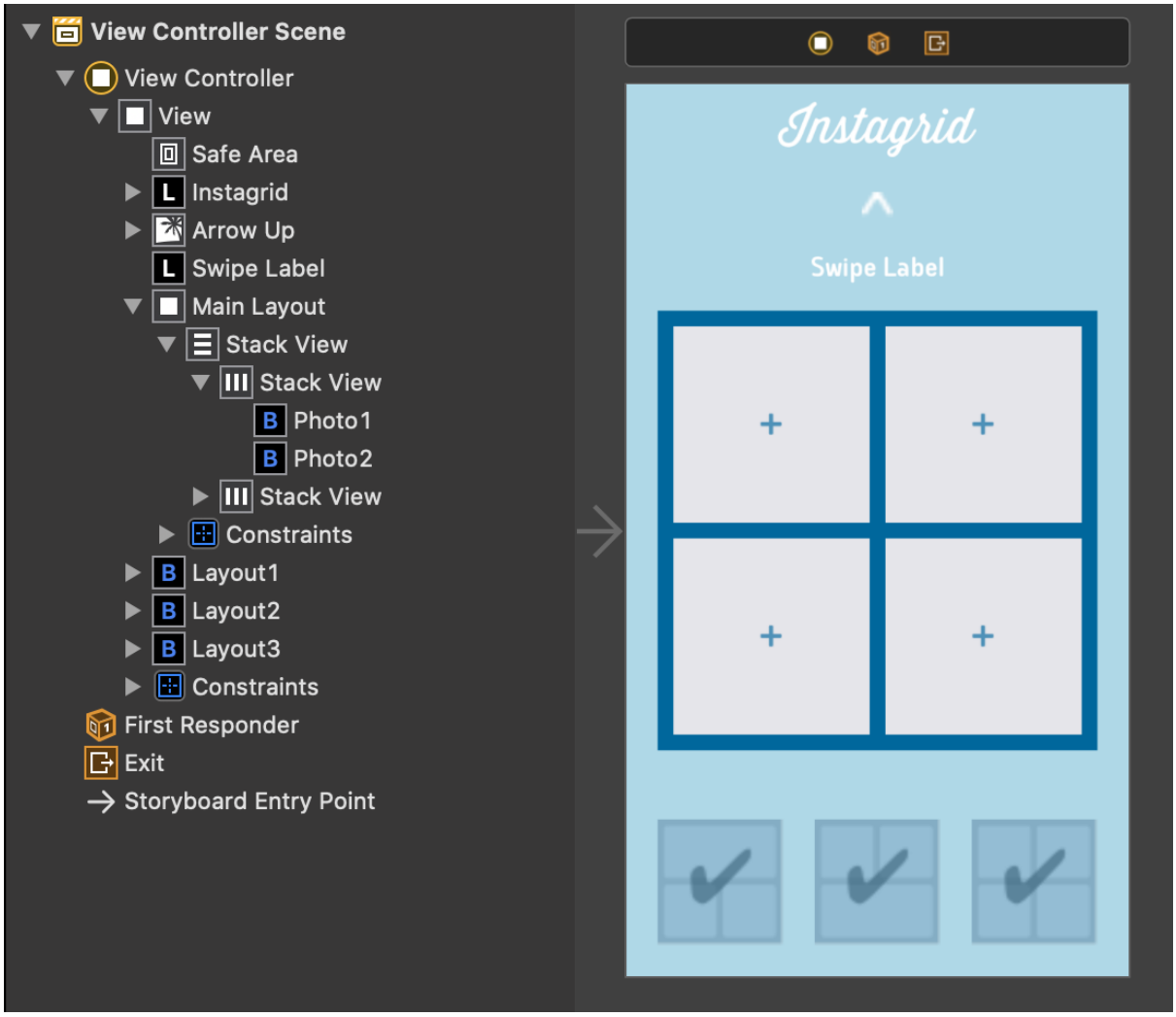

Outlets
Properties
Actions
Lifecycle Methods
Photo Logic
Swipe Logic
Layouts
UIImage Picker Controller

- Outlets : The connection between the views of the storyboard and the controller.
- Properties : The properties declared in the file.
- Actions : The connection between the actions of the screen and the controller.
- Lifecycle methods : This part takes care of the override methods used such as :
  - viewDidLoad, that handle the settings of the User Interface when the application is launched.
  - viewWillTransitionTo, that handle the settings when the user change the device orientation.
- Photo Logic : This part handle the users interaction with the photos and layouts. This part shows a picker controller with the library of the users phone, or change the layout of the collage in the middle of the screen according to the user's choice.
- Swipe Logic : This part handle the pan gesture on the screen. It creates a pan gesture recognizer, add it to the view with the right direction ( up when the device is in portrait mode, and left when the device is in landscape ) , and then show the share controller with the user's favorites applications.
  It also animate the layout out of the view when the gesture is ended. The layout gets back when the share controller is hidden.
- Layouts : This part hide or show the different layouts and selected image when the user change from one layout to another.
- UIImage Picker Controller : This part is an extension to make sure all the methods inside are referred to the according delegate. Here is implement the UIImagePickerControllerDelegate and UINavigationControllerDelegate. The only method implement is called when the user has finished to pick his image

from the library. The method place the picture at the according place in the layout.

## Storyboard Design






The design was made according to the guidelines of OpenClassrooms Project 4.

The different elements explained from top to bottom of the screen.

- Instagrid Label : This label has a custom font ( ThirstySoftRegular provided by OpenClassrooms ) and is 31 point.
- Arrow : This is image view, and change according to the orientation, ( Arrow Up in portrait, Arrow left in landscape ). Its a square of 20points.
- Swipe Label : This label has a custom font ( Delm-Medium provided by OpenClassrooms ) and is 26 points.
- Main Layout : This is a complex view with a vertical Stack Views that embed two horizontal stack views, with

one or two buttons.
The stacks has a « Fill Equally » distribution and a spacing of 10points.
Each button of the stack view has a different tag in order to differentiate them.

- The last elements of this UI are 3 buttons .
  Each button has a different tag in order to differentiate them.

# Discussion

## Difficulties encountered

I never used the storyboard before, and use this in a complete project was completely new for me. I took some time to understand how the constraints work with each other. Especially in landscape mode.

This passed, the difficulty was to create the main layout with simplicity and reusability. First I created this layout without stack views, and it was very verbose ( programmatically ) and repetitive. The pictures wasn't placed at the right position, and some blanks spaces existed.

The Stack View solution appears to be the best one, fast an reliable.

I created first a complex custom view to handle the pictures, it has some extra settings that was not very useful in this case. Then the solution was to used native buttons, and it was first hard to understand how to place and resize the pictures inside them.

## Areas for improvements

This project is great to familiarize with the Xcode environment, the Swift language and a bit with the MVC architecture.

In order to follow a constant way of improvement, the application can contain some other features :

- The ability to allow the user to change the background color of the main layout.
- More and more layouts…
- Let the user change the position of a picture by holding and drawing one.
- And many more if we start thinking about it.