MVC Research

There are many architectural patterns out there, but Apple's recommended weapon of choice is the MVC architectural pattern, otherwise known as the model view controller. The MVC is made up of three objects that work collectively together.

The Model in the architectural pattern is where one can find data consisting of parsers, managers, persistence model objects, data sources & delegates, constants, network code, helpers and extensions, and abstraction layers & classes. At its core , the model encompasses data and notifies the controller when data is needed to be ultimately passed back to the view. There are also a lot more classes and objects , but the ones stated previously are the most common. It is worth mentioning that the model also never interacts with the view directly and always receives updates and notifies the controller exclusively.

The Controller in the architectural pattern acts as the mediator between the model and view if needed , but interacts directly with the user via updates. It is the least reusable part of the model and acts accordingly to domain specific rules. The controller acts as the brains of the model as it determines what is the next step or rather what are the available next steps one should take in the given processes. Classes in the controller typically decide on the next user screen, background refreshes, what's next to access, and etc..Its primary job is to mitigate between the user interface & model , mitigate user inputs, and trigger data loads.

The View in the architectural pattern is what most people are most likely familiar with. It directly deals with the face of the app, user interaction, and overall the business logic of the app. It typically consists of classes from the UIkit, UIview subclasses, and core graphics/animations. The view as a whole interacts with the controller which can then return the update needed or retrieve data from the model to be passed back to the view/user.