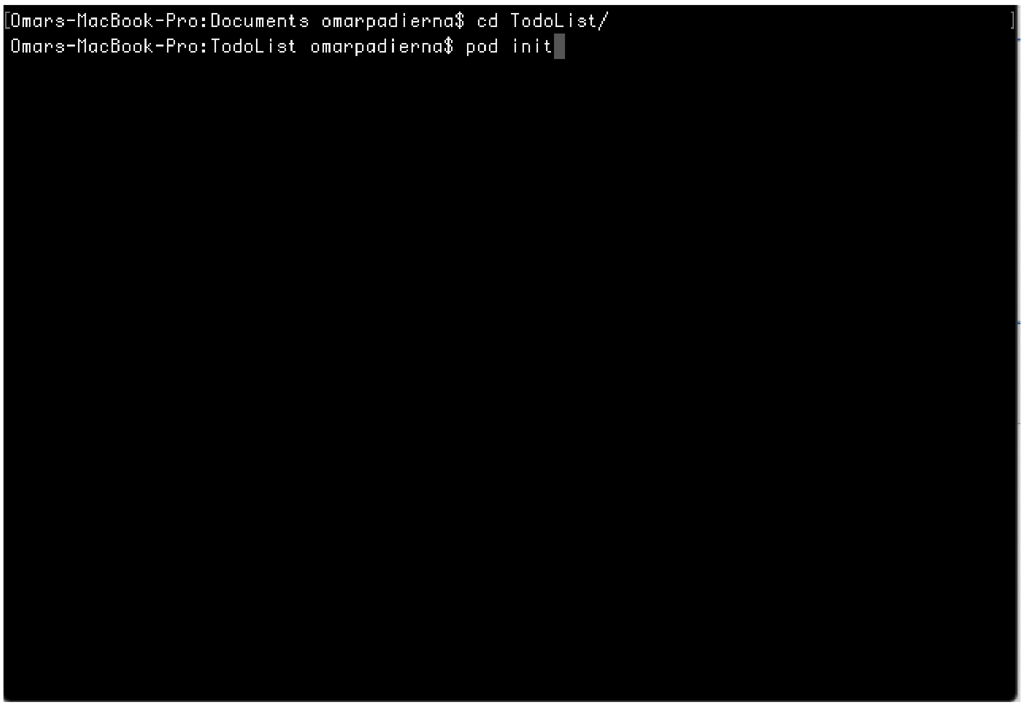
NorthOne coding challenge Readme

Running the project

The project uses Cocoapods to fetch Firebase and calendar dependencies. Cocoapods can be installded using Homebrew

In the terminal, cd into the source directory and run pod init

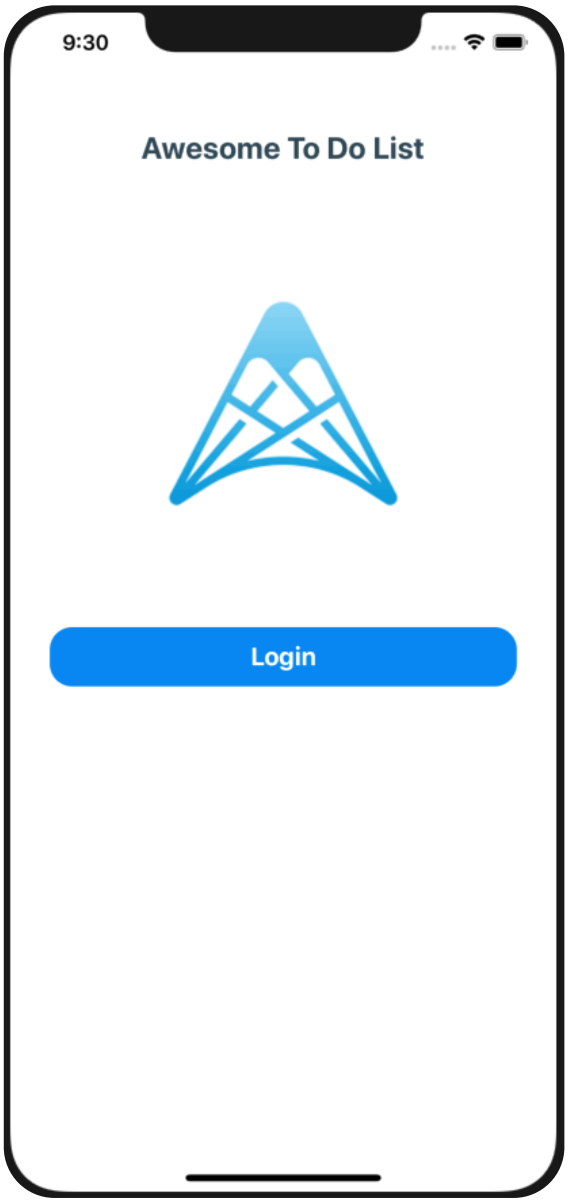


```
[Omars-MacBook-Pro:Documents omarpadierna$ cd TodoList/
 Omars-MacBook-Pro:TodoList omarpadierna$ pod init
```

Open the project's workspace in Xcode and run

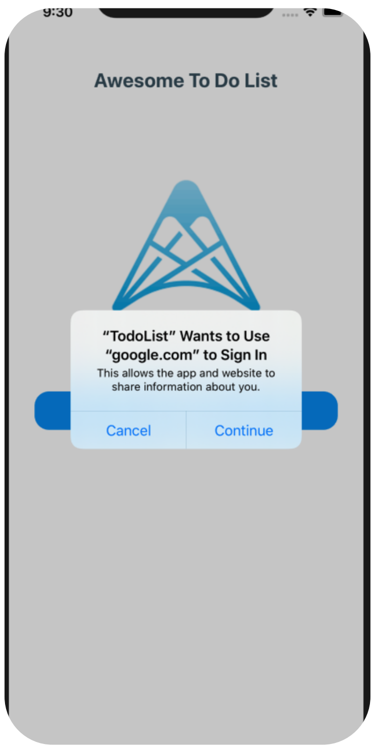
TodoList.xcworkspace

Architecture

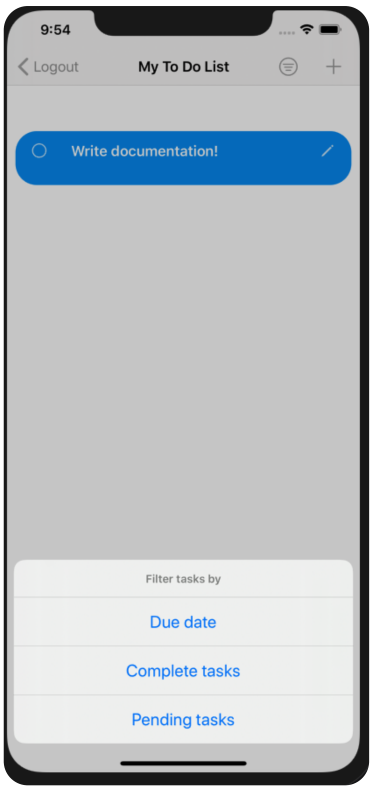


Given the small size oft the project, the architecture used for this challenge was MVC

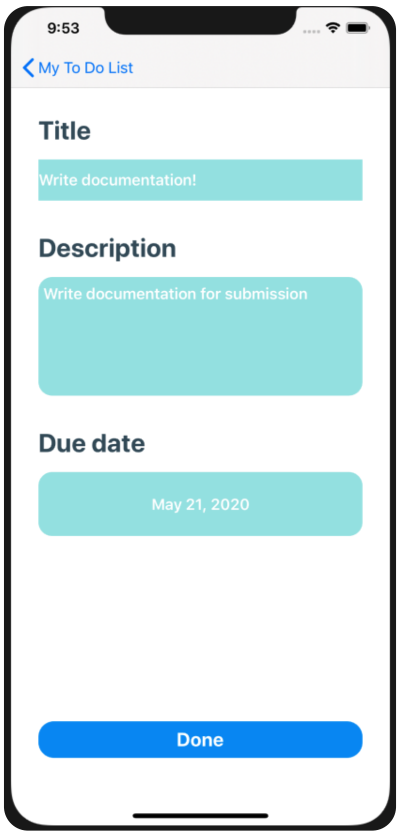
Features


9:30
Awesome To Do List
"TodoList" Wants to Use "google.com" to Sign In
This allows the app and website to share information about you.
Cancel
Continue

Google login via Firebase

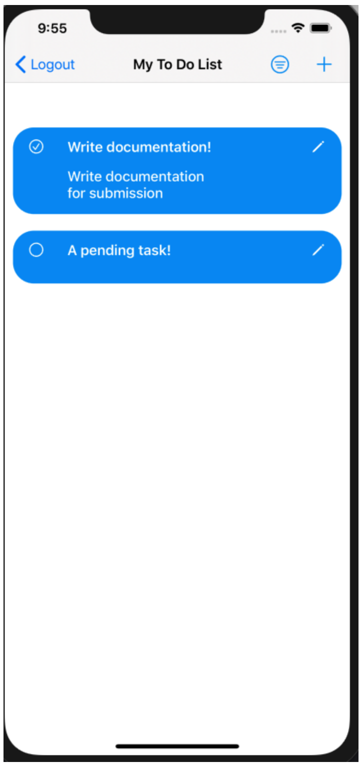


Filtering tasks by:
Due date
Status

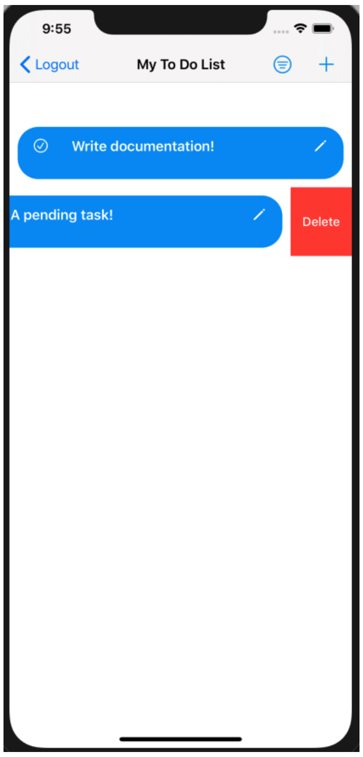


Creating/Editing tasks

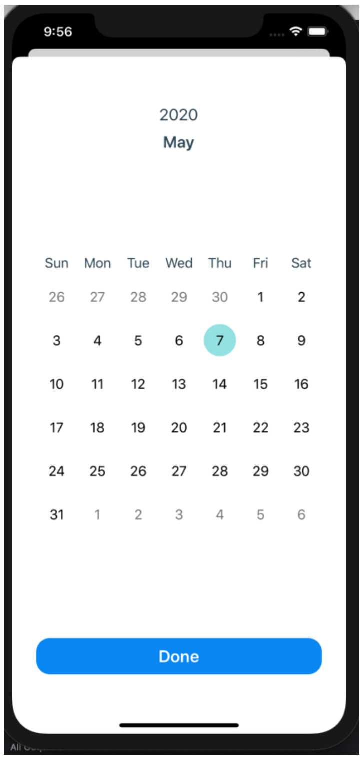
Features



Updating status Collapsible rows to display/hide description

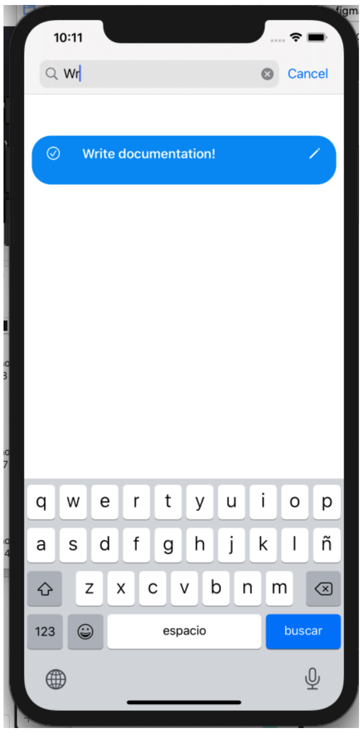


Task deletion



Calendar view for due date filtering and due date selection

Features



Task search

Patterns

The app updates the status and content of each task using Firestore

To achieve this a FirebaseManager was implemented. It follows a facade pattern that contains the update, save, delete and create methods

The FirebaseManager is also in charge of initializing the Firebase SDK.

The Facade is also type safe and encapsulated. It is instantiated as a singleton.

Future work

The TaskListViewController is quickly becoming a massive view controller

To combat this problem, 2 solutions are proposed:

1) Refactor out the controller’s work (particularly when dealing with filtering) by creating embedded view controllers

2) Change app’s architecture into MVVM