

Project Name: Daily List

Members: Talha Ather and Jacob Davis

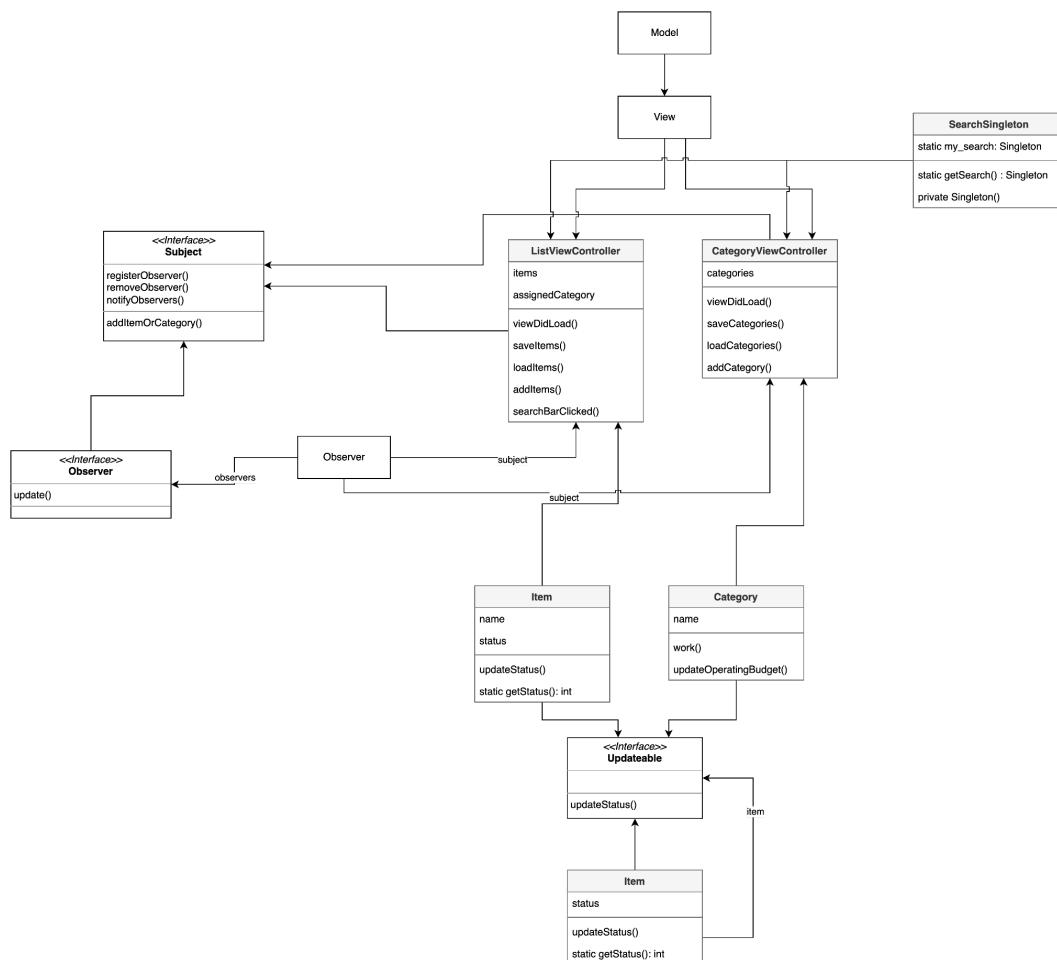
Semester Project for CSCI 4448

State of System Statement

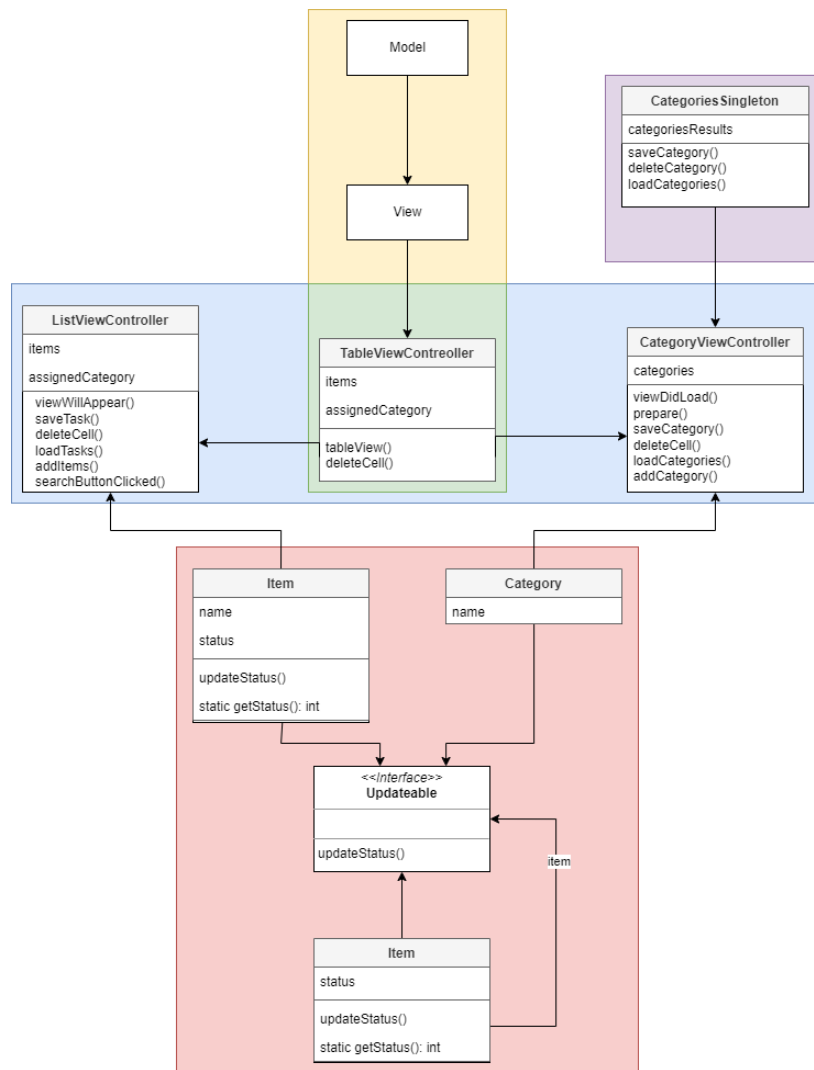
Currently, this app has implemented all the features that we planned to implement in Project 5. The app allows users to add new lists to keep track of what they need to do and add items to each list. These items can also be checked off or fully deleted, and if a list is no longer needed the end user can also delete the list. Finally, within each list there exists search functionality that allows users to search for different items that contain the word that they are searching for. While we were able to implement every intended feature and avoid scope creep when planning for our project, the actual implementation did change. These were notable in our use of different patterns than were originally intended and the implementation of a new class. These changes can be seen in the next section where our original and current class diagrams are provided.

Class Diagram & Comparison

Project 5 UML Diagram:



Final Class Diagram:



Comparison:

The largest difference is that there is no Observer pattern for notifications. During development, we decided that it did not work very well, and opted instead to use the Template pattern, which is indicated above by the area highlighted in blue. The MVC, Delegate, and Singleton patterns look the same and are shown respectively by the yellow, red, and purple backgrounds. A few classes have modified functions to them, as how data storage and displaying in Swift was not completely apparent until development began.

Third-Party Code vs. Original Code

There are two included packages that include third-party code. Otherwise, it is all original. The first package is the primary framework for SwiftUI iOS, and is needed to make any iOS applications function. A link to the package and its tutorial is displayed below:

<https://www.mongodb.com/docs/atlas/app-services/tutorial/swiftui/>

The other third-party software that was used is the iOS Swipe Cell Kit. This allows for users to swipe a list or task to the left, from where it can be deleted. This was implemented with the sole purpose of enabling this swiping and deleting process. A link to the repository is below:

<https://github.com/SwipeCellKit/SwipeCellKit>

Overall OOAD Process

List three design process elements or issues (positive or negative) that your team experiences in your analysis and design of the OO semester project:

1. iOS platforms can use Realm Database for easy storage of local data, which went smoother than it would have if developing this for any desktop operating systems. On top of this, storing everything locally meant we could avoid dealing with cloud storage.
2. Using an MVC allows us to easily update the screen when a user tries adding or deleting something. It allows a nice chain of reaction where an input can be received, processed, updated in the database, and then the display can be reloaded to display the new data.
3. While choosing iOS to develop on was a valuable resource sometimes, development was strictly limited to macOS devices. This was non-problematic until one group member's Mac died, and development was considerably slowed as a result of this.