

CS290.02

Homework 3 - Recipe App v2 with MVVM and DataStore

Let's apply some architecture to your Recipe App. The lectures this week ... and the associated resources ... will be Very Helpful in guiding you through this process.

1. Create a RecipeStore and initialize it with our PreviewData.
2. Create ViewModels for your listView and detailView and move functionality and state into those ViewModels. This effort would include:
 - Renaming any action functions within button taps to have a more generic name in the view and be processed in the viewModel
 - Looking for other opportunities to move sorting or filtering to the viewModel
 - Subscribing to changes in RecipeStore data
 - Include a loading State pattern in your RecipeDetail view and viewModel
3. Confirm that all the above is functioning after the move.
4. Include a searchable modifier on the list, with searching by recipe name and recipe tags. You'll want to make sure you normalize both your searchable string and your search text ... most likely by making everything lowercase.
5. Please provide functionality so that the app user can scale the recipe to feed a larger number of people. The user would pick a scale value and you would adjust the displayed components amounts accordingly¹. To accomplish this, include a Picker on the Edit modal that will allow the user to select a number to scale the recipe by. The Picker options should be 1, 1 1/4, 1 1/2, 1 3/4, 2. Also include a navigationBar button in the upper right of the modal called "Done" that will dismiss the modal. No Cancel button necessary (this time).

You do **not** have to worry about persisting this scale value if the user navigates away from this particular recipe.

Picker Notes: Documentation is here: <https://developer.apple.com/documentation/swiftui/picker>. You probably want to create an enum with those five named cases above and use that enum to power the picker. That enum should provide a rawValue that is a Double (the number you are scaling by (1, 1.25, 1.5, etc.) but also provide you a computed property called localizedDescription that returns a string so you can show the fraction (as a string), not some decimal number.

At the end of this exercise, you will likely have these files (among other standard App files):

- A RecipeStore Object
- A RecipeList SwiftUI View - with nicely structured view code
- A RecipeList ViewModel
- A RecipeDetail SwiftUI View - with nicely structured view code
- A RecipeDetail ViewModel
- A RecipeForm SwiftUI View for the edit modal, but it can use the existing RecipeDetail ViewModel
- And your Recipe models of course.

¹ Pretentious foodies like Nathan Myhrvold, former Microsoft CTO and current modernist cooking laboratory creator, would tell you that you should not scale aromatics like onions and ginger at the same rate as other ingredients; their pungent nature scales more efficiently. We will ignore such mystical concerns. Important life hack unrelated to this assignment though: you should double the amount of sauce in any recipe you see. Who doesn't love more sauce?