

Final Project - README

Purpose: For this project, I proposed making a fitness app. The App serves several purposes. First and foremost, an IFBB Pro colleague of mine asked me if I could create a rough model design of a fitness app for a business demo (you will see his business logo, “Team Kawasaki” on the Home Screen of the app). Second, this app is something that I myself would use. Clients looking to follow a gym-fitness routine that targets specific muscle groups and provides a detailed list of exercises with sets and reps, instructions, and even pictures will reap a great benefit from this app. The pictures and “Note” functions of the app were not in the original proposal, however were something I wanted to include afterwards.

User Manual: When launching the app, clients will see a home screen with the business name and a navigation link. A version is displayed at the very bottom for future use. When the navigation link is pressed, the client will be taken to a “Muscle Group” view which lists different muscle groups (ie chest, back, shoulders, etc). When the client selects a muscle group, they are navigated to an “Exercise Routine” view with a scrollable list of exercises. If an exercise is selected, the client is brought to a “Details” page with instructions, reps/sets, a picture, and a note area (textEditor) where comments can be saved for later review. A singular back button is present throughout the navigation and may take the user back to the previous back, or if held down, all the way back.

Implementation: The app works using xCode in SwiftUI. There are two major swift classes. MainMenu and MockData. The decision was made to include all the code in MainMenu and use “MARK breaks” to separate each view (Main Menu, Muscle Groups, Exercises, Details, Preview). There is a single NavigationView in Main Menu, while the rest of the views use Lists with an embedded Navigation Links to subsequent views. All data is pulled from the MockData class which has included structures. This data class is also broken up using MARK breaks for management ease.

Bugs: Though the “note” function was not a requirement of the project, I wanted to include it anyway. It occasionally appears to not save data (or will load empty data) if the user goes back to ExerciseView and then back to Details. However, if the user goes back to a different view or closes and relaunches the app, it does appear to save the data. Since this was not a requirement, I am comfortable leaving it in the project. *I don't believe this will lose any points since it was not in the original proposal.*

Future Development: Ideally, I would like to learn how to connect this to a server for remote management of the workouts and be able to push updates to the app each week for new exercises. I would also like to have a login feature so this could be a membership based app.

Comments: The most difficult part of this app, believe it or not, was getting the background color in each area of the app. TableView would remain white to no avail. Using .onAppear and UITableView.appearance was able to resolve this issue. Additionally, getting an image to load in a Hashable struct took a little thinking, but a workaround was found. Throughout this project I was able to learn SwiftUI from scratch, ZStacks, VStacks, Spacers, and different formatting functions.