Lifting Log

iOS Application Built For Weight Lifters

Joe Melito and Neil Kalanish

Introductions

- Joe Melito
  - Graduated from Cleveland State University in 2019.
    - Bachelor’s of Engineering in Computer Engineering.
    - Minors in Mathematics & Computer Science.
  - Currently Working as a Software Engineer at Progressive Insurance.
  - Currently Pursuing Master’s of Engineering in Software Engineering.
- Neil Kalanish
  - Graduated from Cleveland State University in 2019.
    - Bachelor’s of Engineering in Computer Engineering.
    - Minor in Computer Science.
  - Currently Working as a Project Manager at Hyland Software.
  - Currently Pursuing Master’s of Engineering in Software Engineering.

Our Application

- We built an iOS application meant for bodybuilders & powerlifters who want to track their workouts.
  - Our main focus was allowing our customers to quickly enter and view data without any ‘bloat’ that is typically found within other workout applications.
- Our application was created using the following technologies:
  - Apple’s Programming Language: Swift
  - Apple’s IDE: Xcode
  - Google’s Database Engine: Firebase
- Lifting Log was architected using MVC.
- Our methodology used was Waterfall.
- Our testing practices focused on manual testing.

Goals

- To obtain a better understanding of Apple’s Developer Tools and Google’s Firebase Database.
- Understand the common practices used when building a mobile application.
- Utilize the knowledge acquired from this class to help us create a piece of software.

Lifting Log’s Feature Set

- User account login, creation, & password reset.
- Submissions of workout data which include:
  - Workout Type
  - Weight Lifted
  - Amount of Reps & Sets
  - Additional Custom Comments
- Ability to view previous workout data
- Ability to compare powerlifting strength against other lifters with Wilk’s Calculation.
- Ability to calculate max weight lifted for a single rep with 1RM Calculation.

Components

Views

- Our User Interfaces Include:
  - Login & Account Creation Screens
  - Main Dash Board
  - New Workout
  - Past Workout
  - Detail Tables
  - Wilks & 1 RM Calculators
- Our views will never know if the information was correct. Their functionality only includes displaying errors or advancing to the next or previous view.

Components

View Extension

- In addition to our user interface views, we built front end methods that help determine:
  - Which alerts should be shown to the user.
    - Submission, Cancelation, Errors, etc.
  - Which view to navigate to.
  - Other interactions the user can do without interfaces such as:
    - Views auto adjusting when keyboard is present.
    - Giving the users the ability to hide or show the keyboard on certain screens.

Components

Structures

- These are the objects we created which pass data around within our application.
- Workouts Structure:
  - Date
  - Workout Type
  - Weight
  - Reps
  - Sets
  - Comments
- AuthResponse Structure:
  - SuccessfulSignin
  - Error

Components

Services

- These are the comments that handle our calculations & talking to our backend database:
  - Calculation Service
  - Database Service
  - Authentication Service

Future Enhancements

- Give user’s the ability to fully use Lifting Log on any Apple Device.
- Implement unit tests.
- Update our UI / UX design to something more modern.
- Allow for user’s to login with Apple ID, Gmail, Facebook, etc.
- Implement 2FA for user accounts.
- Give user’s the ability to filter workouts in the previous workouts view.

Any questions?