SOFTWARE REQUIREMENTS SPECIFICATION

Lifting Log

Joe Melito (2571661) & Neil Kalanish (2651086)

1.0 Introduction

Many people within the lifting community still rely solely on pen and paper to log their workouts. We aim to eliminate that by creating a cross platform iOS mobile application that allows our users to swiftly log, view, and plan their workout sessions. Our application will primarily focus on lifting; which gives us the ability to remove the bloat that other workout apps include.

1.1 Goals and objectives

- To allow our users to signup and create an account using firebase authentication.
- Seem-less UI design that eliminates unnecessary bloat that other workout apps have. We want our users to be able to quickly and easily track and record their weight lifting workout sessions.
- Using Google Firebase to security store all our user's private information.
- Allow cross platform availably with other Apple Products. Such as, MacBooks, iPads, and the Apple Watch.

1.2 Statement of scope

Our software will allow users to input their workout sessions data. Which will then be stored in a database. Users can then view their previous workout sessions within our app. They can also rely on our application to handling any calculations for their workouts such as Wilks and 1RM.

1.3 Software context

Our software's product placement is within the workout & health community. The main feature is to allow users to track, plan, and view their weight lifting workout sessions. We will solely be focusing on weight lifting as we want to remove any unnecessary 'workout bloat' from our application and eliminate the dependency weight lifters have to recording workouts on pen and paper.

1.4 Major constraints

Since our software is being developed for a 16 week course the major constraint is time. The team consists of two developers who are working full time within the software development profession. So learning Apple's programming language, Swift, and Google's Firebase database engine will be a challenge.

2.0 Usage scenario

Lifting Log is a mobile application that will be used by weight lifters to help track, plan, and view their workout data. It will also calculate 1RM and Wilks for a better understanding of their workout sessions.

2.1 User profiles

User profiles will either be created by providing a unique username and password. Users will also be able to login and view their data using Google's Firebase authentication system.

2.2 Use-cases

Users will be able to track, plan, and view their workout sessions within our application. Such as recording the amount of sets, reps, and duration of their workout. They can also plan workouts ahead of time by setting goals. We will also provide the ability to calculate 1RM and Wilks to give users a better insight of their workout data.

3.0 Application Architecture

Lifting Log will be engineering from the ground up using Object-Oriented Programming techniques.

3.1 Data Description

Lifting log will have several data objects. These objects will help store user information so that workout data can be recorded and viewed. See section 3.1.1 for further details.

3.1.1 Data objects

Data objects within our application include:

- User Structure:
    - First Name
    - Last Name
    - Date of Birth
    - Gender
    - Username / Email Address
    - Password
    - Workout Data
- Workout Structure:
    - Workout Type
    - Amount of Sets & Reps
    - Weight Lifted
    - Additional Comments / Details (Custom User Input)

- Workout Calculations Structure:
    - Max Dead Lift
    - Max Bench
    - Max Squats

3.1.2 Relationships

The relationships with our data will be done by creating different micro-services within our code. This way it keeps our code from being too coupled and allows our backed services to be handled separately from our front end. Our micro-services will include:

- Calculating workout data
- Passing data too and from our database
- Handling which views will be shown within our GUI

4.0 Functional Model and Description

Below are the listed requirements for our Lifting Log Application.

4.1. Description of Major Functions

Please See Below:

4.1.1 User Login - Back End

- Allowing for users credentials to be passed security to our back end database. These user inputs include a username and password. Planned integration with Google's Firebase Authentication.

4.1.2 User Login - Front End

- Designing a GUI which contains two textfields; one for username and the other for password. Two buttons will also be created to allow users to navigate to the main dashboard (After inputting their credentials) or to the account / new user creation screen.

4.1.3 User Account Creation - Back End

- Allowing for users to set up an account and have their data securely passed and stored within our back end database. The fields that we will be passing and storing include: first name, last name, username / email, password, and date of birth.

4.1.4 User Account Creation - Front End

- Designing a GUI which contains 3 textfields: first name, last name, username / email. Also a date picker for date of birth and then two buttons which will allow the user to either confirm or cancel the account creation

4.1.5 User Main Dashboard

- Design a GUI interface that allows the user to navigation to the following pages: Create workout, Start Workout, View Past Workout, Wilks Calculator, 1RM Calculator, Settings.

- The design of this will be created using a UI Table View Controller within Swift.

4.1.8 User Settings

- A basic GUI that will allow the user to change settings within the app such as light or dark mode.

4.1.9 User Workout Creation - Back End

- Stores all user workout data within our database. Can be passed back and forth. Data includes: Workout type, amount of sets, amount of reps, weight, and any additional comments the user may want add.

4.2.0 User Workout Creation - Front End

- A GUI that contains textfields to allow the user to enter in their workout type, amount of reps and sets, weight and any additional comments. We will also have two buttons. One to store the data in our database and the other to cancel the workout.

4.2.1 User View Past Workout - Back End

- All data within the “User Workout Creation” view will be sent to our database upon confirmation. The View Past workout will filter out previous workout data from our database with a simple database call.

4.2.2 User View Past Workout - Front End

- User will be prompted to select the date range they want to view their workouts from using a data picker. The corresponding data will be retrieve and displayed in a table. Table headers include: workout type, amount of reps, amount of sets, weight, and comments.

4.2.3 User Wilks Calculation - Front End

- GUI interface with text fields to allow the user to enter in the variables needed: gender, weight, max dead lift, max bench, max squat and an output. Buttons will also be needed to confirm or cancel the calculation

4.2.3 User Wilks Calculation - Back End

- All user data is entered and then a calculation occurs. All data is stored in the database and will reproduce the Wilks output.

4.2.3 User 1RM Calculation - Back End

- GUI interface with text fields to allow the user to enter in the variables needed: weight, reps and an output. Buttons will also be needed to confirm or cancel the calculation

4.2.3 User 1RM Calculation - Front End

- All user data is entered and then a calculation occurs. All data is stored in the database and will reproduce the 1RM output.

4.2 Software Interface Description

Lifting Log will interface with other Apple devices, Google's Firebase, and Apple's industry standard GUI interface, Swift Storyboard.

4.2.1 External machine interfaces

Our application will be interfacing with all iOS devices running iOS 16.0 or higher. We hope to also allow our application to interface with other Apple Devices running different operating systems; such as, MacOS, iPadOS, and WatchOS.

4.2.2 External system interfaces

Our application will be interfacing externally with Google's Firebase database engine. We will be storing all user information within a custom build firebase database. While also using Firebase authentication to allow our users too create accounts and login to our application.

4.2.3 Human interface / GUI

Developing within Apple's ecosystem limits us to using Apple's IDE, Xcode, and Apple's programming language, Swift. Due to these limitations our human interface or GUI will be created using Swifts building in Storyboard interface. This allows us to create an industry standard user interface for not only mobile devices, but any device found within Apple's product line.

5.0 Restrictions, Limitations, and Constraints

As part time students with full time jobs working within the software development profession; our limitations include learning iOS development and Google's Firebase database engine. We will be developing this project completely from scratch while also teaching ourselves Swift, Xcode, & Firebase. We would also like to plan on having our application cross platform with other Apple device's, but given our lack of knowledge and timeline for the project this may not be a feasible commitment.