**Software Implementation and Testing Document**

**For**

**Group 2**

Version 1.0

**Authors**:

Sydney McGinnis

Sophie Pavia

Henry Gilbert

Alejandro Serrano

Jack Garthwaite

# 

# **1.** **Programming Languages (5 points)**

*Currently the majority of our project built using Swift and in XCode. We chose this platform because it is the easiest way to create an iOS app and Swift is the primary language used for iOS app creation. We are also in the process of integrating a php api that connects us to a SQL data base.*

# **2.** **Platforms, APIs, Databases, and other technologies used (5 points)**

*List all the platforms, APIs, Databases, and any other technologies you use in your project and where you use them (in what components of your project).*

*We currently use XCode as our primary platform. We also have a Sign In with Google API used in the login screen of the app. This API lets the user login if they have a Google account. Before integrating this into the project, we registered our app with Google so that we would have the proper Client ID to link the app and the API. Finally, we are also implementing a self-created API to connect xcode to a data base.*

# **3.** **Execution-based Functional Testing (10 points)**

*Describe how/if you performed functional testing for your project (i.e., tested for the* ***functional requirements*** *listed in your RD).*

In this increment, we did not worry about testing anything too rigorously, but we still took a unit testing approach where we each individually tested any new features we added to the app. Also, we were pushing and pulling out commits often, and consistently interacting with the app to fix any errors we came across. We still need to complete component and system testing which will be done in the next increment. Also while doing unit testing, if our individual feature did not work correctly, we did not push it to GitHub until it was fully working.

# **4.** **Execution-based Non-Functional Testing (10 points)**

*Describe how/if you performed non-functional testing for your project (i.e., tested for the* ***non-functional requirements*** *listed in your RD).*

We tested the non-functional requirements by closing and opening the app and making sure the user data remained unchanged. We are in the process of making sure the interface is user friendly by having back buttons and scrolling views of workouts. We tested to make sure the user stayed logged into the app through Google as long as the app was downloaded on the phone. We still need to do a few more tests to make sure all of the user interface works correctly.

# **5.** **Non-Execution-based Testing (10 points)**

*Describe how/if you performed non-execution-based testing (such as code reviews/inspections/walkthroughs).*

Each week we met in person or through Zoom to go over the code we had written previously. As new changes were pushed to GitHub we fixed any code that gave us errors. For the most part, we would also go over our changes each week, and show our code to each other. Oftentimes, the code is simple enough to understand but when something complex is implemented, or someone doesn’t understand something, the person responsible for the task walks the others through what was done. However, a deeper inspection is needed to ensure that all variables in each view controller are initialized and stored properly. This way once we have a working database, no ambiguity arises.