

Software Implementation and Testing Document

For

Group 3

Version 1.0

Authors:

Cathy Yue

Joao Valente

Damon Akins

Andrew Perez-Napan

Nikolas Hernandez

1. Programming Languages (5 points)

List the programming languages use in your project, where you use them (what components of your project) and your reason for choosing them (whatever that may be).

For this entire project, we decided that the Java programming language would be best, as we will be using the Android Studio IDE to build our workout application. Android Studio only allows users to program in Java or Kotlin and most of the team members have more experience using Java as the primary language.

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 will be embedding youtube video links within the workouts tab of our application. The user will be able to view the individual exercises for their workout routine that day. Next to those individual workouts, there will be a small viewing window with an embedded video, showing users the proper way to complete that particular exercise. To embed the videos we will use Youtubes' API. We have decided to use *back4app* as the primary database for our app with google's *firebase* database as a backup option.

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).*

For Functional testing we use the built in android emulator, which will run the app exactly how it would be run on a phone, to ensure the app works as intended.

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).*

The app does not take longer than four seconds to load as we have tested that with the android emulator. The other non-functional requirements have not been tested as we have not integrated our data base for the users information. By the end of the next iteration we hope to have this integrated and will be testing these features using the android emulator or on an actual android device.

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

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

The android studio ide has built in error checking for code written. Therefore the program will not be built if there are any errors in the code. Also, we review the code before we push it to Github to ensure that it does not disrupt any other code that has been pushed to the repository.