**Software Implementation and Testing Document**

**For**

**Group 5**

Version 1.0

**Authors**:

Felipe Bergano

Bohdan Kovalyus

Mason Metcalf

Michael Nguyen

John Washer

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

We will be using Java for backend development/object-oriented programming and XML for frontend/UI development, as these are the native languages used by Android Studio.

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

* Android Studio as the IDE
* Java for object-oriented class design and data storage
* Android API for account creation, data storage, and UI functionality
* MPAndroidChart for creating a graphic representation of the user's progression in different exercises
* Adobe Creative Studio for UI design
* Firebase for user authentication

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

* Edited lifting goals and ensured new exercise maximums were properly updated.
* Added custom Workouts to the Workout library and monitored their persistence.
* Created a Workout by adding exercises from the user's exercise library.
* Began Workout and ensured the Workout maintained the appropriate values from creation.
* Tested the editing of set weights, reps, and RPE.
* Selected both “Success” and “Failure” for arbitrary sets and ensured this was reflected in the corresponding json file.
* Viewed Progress of the tested Workouts and ensured the data was visible and discernable.
* Tested the dates inputs.
* Made sure no extraneous information was being displayed in any tab.

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

1. Useability: User is greeted with a standard splash screen featured in many apps, where they are then greeted to log in/register to use the app. Once logged in, users are greeted with a basic navigation system with headers that help them discern where information is located. Hence, familiarity helps the user navigate the application. Through integration testing we ensured interactions with APIs such as MPAndroidChart work properly for various states of operation.
2. Reliability & Performance: Tested the responsiveness of buttons, activities, and general time complexities, while understanding the limitations an emulator may impose on such. Also performed range checking on areas requiring user input to ensure a certain value doesn’t crash the program.
3. Storage: Frequently used the File Device Explorer to monitor the creation and modification of files storing user data.
4. Security: Ensured that any locally stored files did not contain any sensitive information.

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

* Reported any bugs in the Discord server and worked cooperatively to resolve them.
* Reviewed each other's code to find faults and make suggestions on areas needing improvement.
* Did code walkthroughs to better understand code functionality and to look for areas of code that may cause issues.