**Software Requirements and Design Document**

**For**

**Group 5**

Version 1.0

**Authors**:

Felipe Bergano

Bohdan Kovalyus

Mason Metcalf

Michael Nguyen

John Washer

# Overview (5 points)

So many mobile fitness apps being catered toward cardio exercise and diet planning results in a cumbersome UI and overwhelming exercise search results, especially for those who are interested in a strength training program. The goal of Strictly Gains is to remedy this by providing a simple UI with all of the necessities catered toward weightlifters of all levels. The lifting in itself is hard enough, so we aim to make logging workouts easy.

# Functional Requirements (10 points)

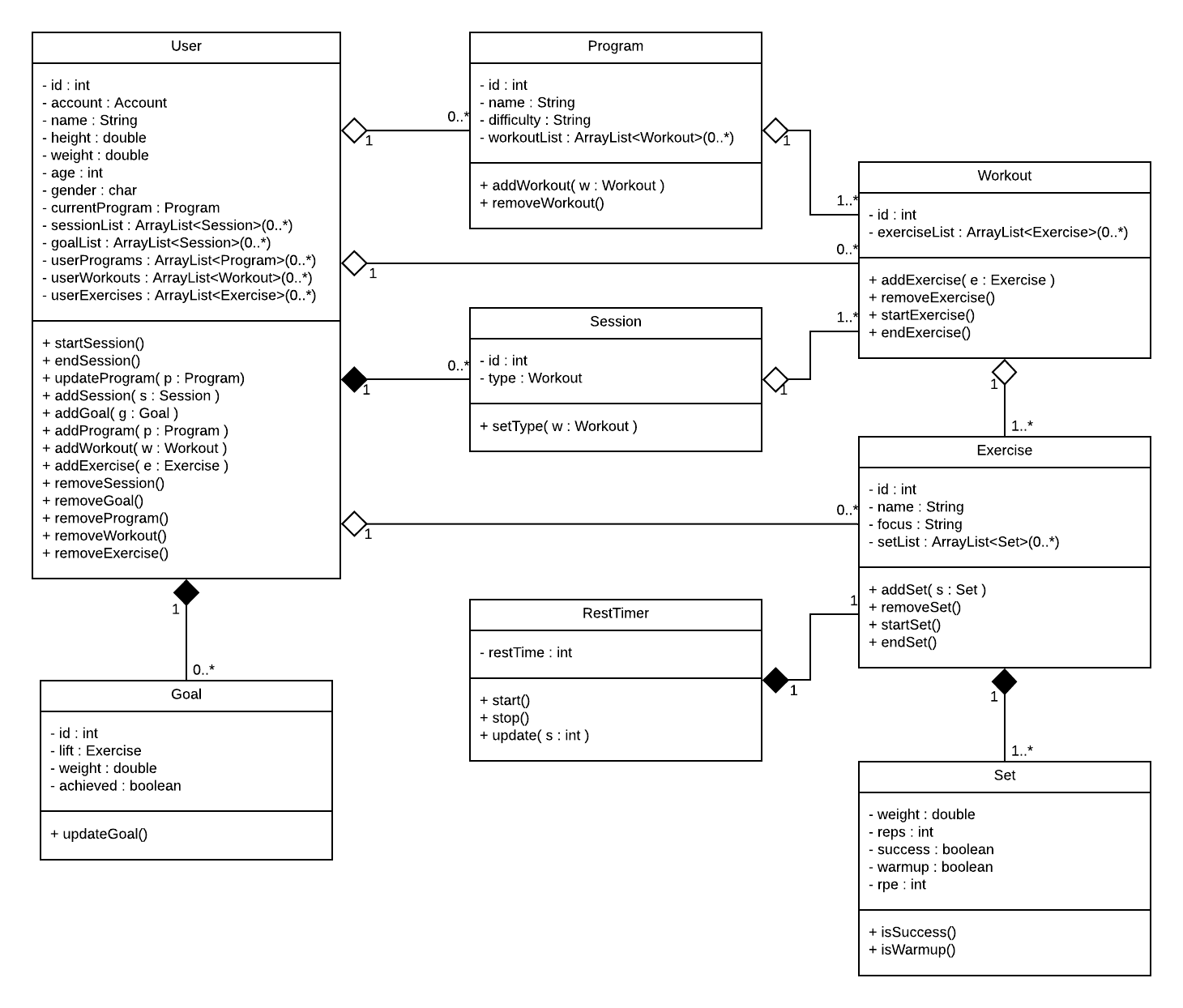
1. **(High)** The user can create a profile that stores their name, height, weight, age, and gender.
2. **(High)** The user can set goals for personal best lifts.
3. **(High)** The user can create their own programming by specifying workouts for specific days.
4. **(High)** The user can create a workout by adding exercises from a premade library.
5. **(High)** The user can control when they start and end their workout session.
6. **(High)** The user can edit the weight of their sets.
7. **(High)** The user can view their lifting progress over time in line graph format.
8. **(Medium)** Warm-up sets will be calculated for the user based on their workset weight.
9. **(Medium)** The user can indicate whether or not a set was failed.
10. **(Medium)** The user can start, end, and edit a rest timer in between sets.
11. **(Medium)** The user can add custom exercises to their workout.
12. **(Medium)** One rep max (1RM) will be calculated, stored, and tracked based on the user's current workset weight.
13. **(Low)** The user can select an option to have their lifts’ weight increase linearly by a user-defined amount.
14. **(Low)** The user can “de-load” their lifts by decreasing the weight by a user-defined amount.
15. **(Low)** A graphic will show the weight plates necessary to reach the lift’s specified weight.

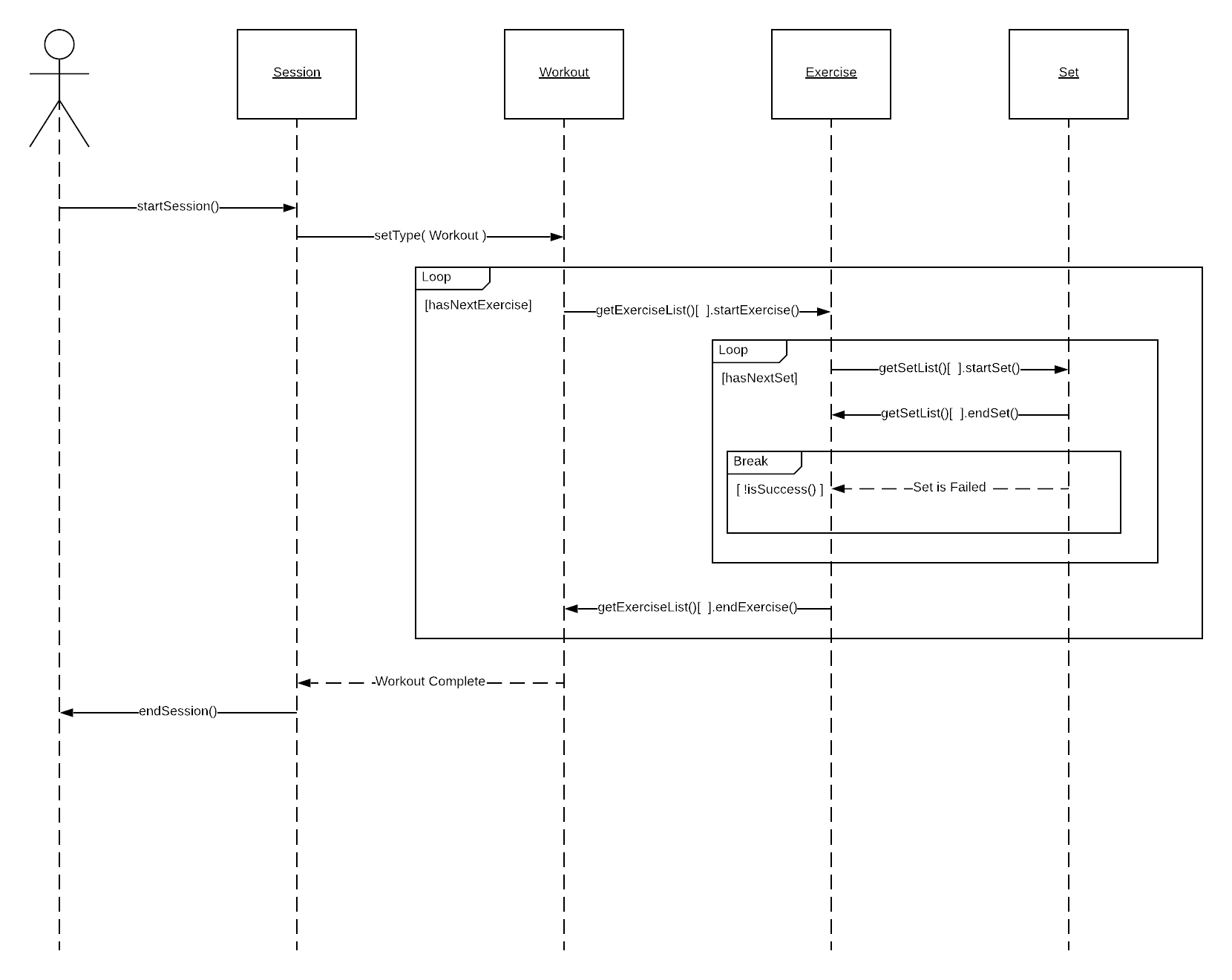
# Non-functional Requirements (10 points)

1. Use-ability: The user should be able to understand the flow of the app easily and use it without needing guidelines.
2. Reliability: The application should be reliable and respond to all user actions.
3. Performance: The application should be responsive with fast load times and no noticeable lag.
4. Storage: User data will persist across multiple app (and workout) sessions.
5. Security: User data will be safely and securely stored locally or using Google Drive.

# Use Case Diagram (10 points)

# Class Diagram and/or Sequence Diagrams (15 points)

**

**

# Operating Environment (5 points)

Our software will be created for the Android OS, supporting versions 5.0 (Lollipop) and higher. The application will be optimized for and tested on the Google Pixel 2 with 5.0” display.

# Assumptions and Dependencies (5 points)

1. Five individuals will be contributing to the project.
2. We will communicate via instant message in Discord daily and have a weekly call to discuss the current state of the project.
3. Our reliance on MPAndroidChart could affect the performance of our “Progress” feature depending on what changes, if any, are made to their API.
4. To implement accounts, we will need to set up authentication through the Google API Console.
5. We will otherwise be implementing all other requirements using the Android API.