Problem Statement and Goals
Software Eng 4G06

Team 2, Parnas' Pals
William Lee
Jared Bentvelsen
Bassel Rezkalla
Yuvraj Randhawa
Dimitri Tsampiras
Matthew McCracken

Table 1: Revision History

Date	Developer(s)	Change
24/09/22	Jared Bentvelsen, Yuvraj Randhawa, Bassel Rezkalla	Initial draft of Problem Statement

1 Problem Statement

1.1 Problem

Working out and exercising is a vital component of any healthy lifestyle, and has been shown to greatly improve both physical and mental well-being. However, many individuals feel reluctant to go to a gym or establish any consistent workout routines simply because they aren't sure what to do. Existing exercise content is mostly expensive Excel spreadsheets or brief, hard-to-follow Tik-Tok clips which are inconvenient to use while you're exercising or in the gym. There does not yet exist an application to track progress and discover new workout content in an easy and effective manner. Workout routines can vary infinitely depending on an athlete's goals and preferences. A workout can be composed of any number of exercises and each of these exercises can be done in any number of variations, with different technical adjustments.

The high level goal of this project is to make working out highly accessible for beginners and to spread new tips and techniques to even the most experienced athletes in a clean and accessible manner.

1.2 Inputs and Outputs

Inputs:

1. A user's favourite workout routines for accomplishing a specific goal (e.g. build strength on bench press) for others to discover
   - (a) General workout description
   - (b) A set of exercises
   - (c) Sets, reps, and weight where applicable
   - (d) Technical comments, pieces of advice
   - (e) Self recorded technique demonstration videos
2. Comments and reviews for other posted workout routines
3. Fitness goals that a user wishes to reach in an optional time frame (e.g. Run 5km in under 25 minutes, or lift 225lbs on bench press)
4. A user's actions during a workout, or data progressing towards a set goal (e.g. Record weight on the bench press, and 'tick off' exercises as they are performed during a session.)

Outputs:

1. Searchable workout content for users to discover programs for specific goals.
2. A method for users to seamlessly track their own progress through a routine or towards their more general goals.
3. Reviews and comments left by other users indicating the quality of a given workout routine.

1.3 Stakeholders

1. Anyone interested in exercising (both beginner and advanced)
2. Personal trainers
3. Fitness Advertisers

1.4 Environment

Supported platforms: iOS/Android mobile application, web application accessible from browser

2 Goals

Goal Name	Explanation
The app should provide a fast, efficient, and easy way to create programs.	Creating a program shouldn't feel like pulling teeth. It should feel more intuitive and convenient than creating a program on an excel spreadsheet - otherwise the value provided by this app decreases.
App provides an accessible way for users to discover programs that are in-line with their goals	A user should be able to search for programs in a simple manner. The programs presented should be very informative in regards to whether or not they work.
Users can use active programs in a way that doesn't intrude or interrupt their workout	Tracking/logging workouts is an essential part to ensuring that the user is on track with their goals. However, it has traditionally been a tedious process that is often neglected. Tracking and logging workouts should be a streamlined process.
Programs should be polymorphic - they can accommodate any style of workout.	Fitness programs can be quite diverse. There's a wide range of ways to configure a program - programs can vary by duration, workout intervals, exercise content, etc. It is important for the program creation process to allow efficient creation of diverse ranges of programs.
App presents data about the user's active program that is insightful and analytical that may not be evident to the user.	Not seeing immediate results can turn people away from sticking with their workout regiment. Displaying insightful user statistics throughout the duration of their program can create a positive feedback loop of hard work vs. rewards. It's important that the statistics presented are not necessarily obvious and evident to the user.

2.1 Stretch Goals

Goals	Explanation
Diet Planning	Users can track their diets and include diet information in their custom workout plans. Some workouts are designed around bulking or cutting calories and so including diet info will improve utility.
Recommended Workouts Algorithm	Users can receive new workouts in a ”Recommended” tab. These workouts will be selected for the user by analyzing their past plan history and plan tags. Popular and new workouts will be recommended to the user. This will improve discoverability and improve user experience and keep them on the app for longer.
History of growth and projection of potential future growth	Users can chart their growth and see an analysis of their future growth if they maintain current trends. This can serve as a motivation to keep up with their regimen.
Social Growth Analytics	Users can view a network of other users who have similar goals and learn from their progress. Users who improved their abilities by following a regimen can act as a guide to others looking to make progress. This feature can improve users' ability to see success with our app.

3 Development Plan

3.1 Team meeting plan

Team meetings will be conducted weekly on non-holiday Mondays at 4:30pm. Further meetings can be scheduled if needed. Meetings will be conducted in person at a McMaster bookable room or library unless specified otherwise.

3.2 Team communication plan

Common team communication and questions will be conducted on **Microsoft Teams**.
Any online meetings will be conducted on **Microsoft Teams**.

3.3 Workflow plan

GitHub will be used for all workflow and version control.

3.3.1 Task Scheduling

The **Github Kanban** project board will be used to plan and assign upcoming and ongoing tasks to team developers.

If a task is large or general, sub-tasks will be created and assigned accordingly. Upcoming project deliverables must include deadlines.

3.3.2 Git Branch Usage

All features, documents and file changes require a named branch describing the change. Branches will follow a similar format to:

$$topic\_or\_type/section/branch\_description$$

3.3.3 Git Commits

Tags will be attached to commits as needed to further display the change description.
Commit Squashing may be used prior to creating pull requests to clean up unnecessary commits.

3.3.4 Git Branch Merging and Pull Requests

The main or centered branch will be protected such that new features and additions will require a pull request in order to be merged.
Pull Requests require a minimum of 2 approvals to be accepted.

3.3.5 Technial Issues

For all technical questions, concerns and issues, the **Github Issues** feature will be used to open an issue or concern accordingly.

3.4 Team Roles

Team Role Descriptions	
Meeting Organizer	William
Front End Expert	Dimitri
Database Experts	Jared, Bassel
UX/UI Expert	Matthew, Dimitri
Networking Protocols Expert	Bassel
Backend Expert	Jared, William
Lead Testers	Yuvi, Bassel
Team Liason	Jared
Backup-Team Liason	William
Git Experts	Yuvi, Matthew
Scrum Leader	William
Mobile Development Expert	Dimitri, Matthew
Latex Expert	Yuvi
Hardware Expert	Bassel

3.5 Coding Style and Standards

The **Google Coding Style** will be used as a standard for all code in the project. The style guide for this can be found here. A language specific linter will be enforced.

3.6 Testing

3.6.1 Unit Testing

Jest will be used for this app's unit testing. Jest offers mocking and code coverage insights which will be useful for unit testing.

3.6.2 Simulation / Integration Testing

TestCafe will be used for this app's integration testing. TestCafe is a cross-browser integration testing framework that will be used to verify proper integration between the app's various components (back-end, front-end, database, etc). TestCafe also offers end-to-end testing which includes automatic simulation of many common web-app components such as forms and buttons.

3.7 Proof of Concept Demonstration Plan

The risks associated with the project are:

1. Implementing a workflow for users to add workouts and for other users to discover those workouts. If this interaction does not work then our application will not fulfill its goals.

2. Building a backend server capable of interfacing with front end clients and storing user information in a secure database. This backend is essential for our application to meet the social networking aspect of our proposed project.

3. Achieving %70 code test coverage. This is a challenge because unit tests alone will not cover most code. View code will need end to end tests for full coverage and this will involve our team working with testcafe which can be time consuming to learn and write.

4. Developing a cohesive UI/UX for our application. All user interactions must be mapped out. We must use group meetings to discuss how each interaction should look and map out user navigation.

5. Using git branches for up to 6 people to do simultaneous development and avoiding pushes to main or unsafe merges. Our group has varying levels of git experience and so we must work together to ensure smooth development.

During the Proof of Concept Demonstration we will prove that these risks have been overcome by showing:

1. A working demonstration of a test user creating a workout and another test user discovering the created workout. Viewers will be walked through the process of creating their own workout and can see how these workouts are then displayed to the general userbase.

2. A demo of a new user signing up for the application and manipulating their own data. Viewers can then see this new user and their custom data in the Database to prove that the backend and server are cohesive with the frontend.

3. End to end tests written in testcafe that simulate the user experience across the application. This will include testing for user sign in, workout creation, and workout discovery.

4. Designs created in figma that describe desired user interactions and portray a cohesive application wide theme.

5. A conflict free git history with even contributions from all members of the team.