## 

## horizontal line



**RepQuest**

09.05.2025

**─**

Dylan Hulon

Austin Jones

Jaylen Cook

Benjamin Johnson

# **Abstract**

RepQuest is an Android-native application that allows users to input and track custom workouts while simultaneously providing for an entertaining experience that incentivizes commitment to the user’s fitness goals. The app allows users to create custom workout templates, log metrics such as sets, reps, weight, or distance in real time, and run a built-in rest timer that persists in the background. To drive consistency, a lightweight gamification layer awards experience points for completing sessions, hitting new personal records (PRs), and maintaining streaks. PRs are automatically detected, and progress can be visualized through graphs generated from user data. By coupling convenient workout tracking with motivational feedback loops, RepQuest aims to foster improved adherence to fitness goals, celebrate incremental progress, and make structured training more engaging and accessible for athletes of any sport at any level of experience.

# **Description**

RepQuest is an Android-native application that provides users with a convenient and entertaining way to create and track custom workouts of various types. The app allows users to create custom workout templates, log metrics such as sets, reps, weight, or distance in real time, as well as track rest periods with a built-in timer that persists in the background. A lightweight gamification layer is applied over the app to incentivize consistency and enhance motivation to reach the user’s fitness goals. Users will receive experience points for various accomplishments such as completing sessions, maintaining workout streaks, and reaching new PRs. The goal of RepQuest is to make structured training more convenient and engaging to better allow for users to reach their fitness goals.

Many people begin training with ambitious goals but soon give up due to low motivation, poor feedback, and friction in tracking their progress. Moreover, many apps do not allow for a convenient way to structure and track a fully customizable workout plan that is not centered solely on either weightlifting or endurance-based sports. Other solutions such as paper notes, spreadsheets, and generic apps are slow, tedious, or distracting to manage mid-workout. As progress towards fitness goals tends to happen slowly, users need immediate, meaningful feedback loops to celebrate small victories and build momentum towards their goals. RepQuest seeks to meet these challenges by combining effortless tracking with playful, goal-oriented game features that promote positive feedback loops and celebrate small victories as well as major milestones.

The primary purpose of RepQuest is to increase workout adherence and enjoyment by transforming training into an experience that is both convenient to track and entertaining. People tend to sustain healthy habits when their effort is visible, progress is recognized in a meaningful way, and rewards align with their goals. Through convenient logging of customized workouts, graphs that show user progress towards their goals, and positive feedback loops, RepQuest can help users achieve more consistent training.

This app targets individuals of all experience levels, whether they are new to fitness, returning trainees seeking to rebuild consistency, or experienced athletes who value structured training and PR tracking. It is intended to be used primarily in the gym during sets, between sets for rest timing, and after workout sessions for reflection and planning. The basic workflow for the app first involves the user inputting their customized workout prior to beginning their training so they can then start a planned workout and flow through the exercises without losing focus. During the workout, the streamlined design of the app will allow the user to log sets/reps/weight or distance/time quickly with minimal taps. Rest periods will be tracked by a reliable, persistent timer that will remain in effect while switching views. At the end of the workout session, the user will be able to see any experience points earned, level-ups/power-ups acquired, PRs detected, and will have the opportunity to view trends in a graphical format so they may see the progress they have made and better plan future workouts.

With RepQuest, users will have access to a multitude of core features that will make training and tracking progress more convenient and entertaining. At the forefront, users will be able to create custom, reusable workout templates with full control over exercise selection as well as metrics such as sets/reps/weight, distance/time, and rest periods. Workouts can be either weight training based, endurance based, or a mix of both. Rest periods will be tracked with a reliable, easy to see timer that continues while the user browses other parts of the app or even while the app is running in the background. While training, users will be able to log their progress through the workout quickly using one-handed gestures with subtle haptic feedback to maximize convenience and minimize cognitive load. New PRs will be automatically recorded without any extra steps from the user. A lightweight gamification layer will award the user experience points for things like completing workouts, maintaining streaks, and hitting PRs which will allow the user to level up their profile and unlock various power-ups. The user will be able to visualize their progress over time with the inclusion of simple graphs that highlight trends, streaks, and recent victories.

RepQuest matters because it addresses the biggest reasons people abandon their fitness goals: inconvenience and fading motivation. Instead of fumbling with paper notes, spreadsheets, or apps intended for specific workout programs, users are able to build truly customized workouts that meet their goals, whether they be focused on weightlifting, endurance, or both. Progress is made visible to the user to provide meaningful training insights and help motivate the user to continue making progress. Also, new PRs are recognized automatically, small wins contribute to streaks, and sessions earn experience points that reinforce consistent training. This combination of tailored convenience and positive feedback provides a frictionless experience that turns slow, incremental improvement into a satisfying progression. By celebrating both small victories and major achievements, RepQuest has the potential to help newcomers stick with their training, returning athletes rebuild their momentum, and experienced lifters stay engaged with their training rather than feeling like a chore, increasing the likelihood that all users reach their fitness goals.

# **Feature List**

## Must Have

1. The user will be able to input custom exercises to create a custom workout template including the ability to:
   1. Set a custom exercise name
   2. Choose a “unit of completion” for the exercise, being either a repetition (rep), distance, or time period
   3. Determine the number of sets
   4. Set the rest period between sets of the exercise
   5. Save the custom workout to be used as a template to track future workout sessions.
2. The user will be able to track their workout in real time with the ability to:
   1. Mark the completion of a set with a single tap
      1. The user will receive haptic feedback upon this gesture
      2. More than one tap will allow the user to decrease the number of reps actually completed
   2. Track their rest period with a rest timer that will automatically begin once a set is marked complete
      1. The timer will continue to run even if the user switches views or the app is running in the background
      2. Upon completion of the timer, the user will hear a chime and receive haptic feedback to notify them the timer is done.
      3. The user will be able to start, pause, resume, and skip rest timers, with optional vibration or sound cues.
   3. Add notes to be saved with the workout.
   4. Record bodyweight to be saved with the workout
   5. Mark the workout as “Finished” at any time during the workout.
3. The user will be able to set default units in the settings menu.
4. The user will be able to earn experience points for completing sessions, maintaining streaks, and achieving PRs.
   1. Experience points will be used to level up and unlock milestone badges that celebrate consistency and progress.
   2. Breaking their own goal will be incentivized to the user by offering a bonus challenge, slightly higher than their last PR.

## Could Have

1. The user may have the ability to play a small mini-game during rest periods between sets. Some examples include:
   1. A small color game where the goal is to turn every tile on a board into the same color. Select a color and tap a tile to turn every connecting tile of the same color to the selected color. Turn the board into the specified goal color to win.
2. The user may have the ability to change the appearance of the app from light mode and dark mode.
   1. They can also unlock new cosmetic themes upon leveling up, ranging from flat colors to background images and patterns.
3. The user may have a customizable avatar.
   1. The user upon leveling up, completing a challenge, or reaching a new personal record can unlock new outfits, accessories, or gadgets to an avatar player. This avatar would likely be purely cosmetic.
4. The app will feature guides for users new to health and exercise. There will be a section with recommended beginner exercises and dieting help.
   1. Some of this information will also be presented as tips at the bottom of the screen after a workout.

## Won’t Have

1. The user can send and receive friend requests to interact with other users of the application
2. The user can share and import exercises and workouts from other users using a code or link.

# **Initial Set of Technologies**

* Platform: Phone, Tablet
* Operating System: Android OS
* IDE: Android Studio
* Languages: Kotlin, Java, SQL
* Frameworks: Jetpack Compose
* Database: SQLite
* UI Design: Figma
* Communication Software: Discord
* Version Control: Git, GitHub, GitHub Desktop
* AI Platform: ChatGPT
  + ChatGPT will be used more for general guidance, planning, and advice rather than writing substantial portions of code. This decision was made to foster the learning experience of the group in working with new technologies on a fairly large project.

# **Backgrounds**

## Dylan Hulon - Role

Familiarity with the Java, Python, SQL, and JavaScript programming languages as well as the JavaFX, React, React-Native, and TailwindCSS frameworks. Development environment experience includes IntelliJ, Visual Studio Code, and Android Studio. Functional knowledge of version control using git and GitHub. Experience with Windows, Android, and Debian based operating systems.

## Austin Jones - Role

Familiarity with Java programming, JavaFX programming, Lua script writing, as well as familiarity with Android operating systems. Minor experience in the physical construction of building desktop computers, specifically Windows OS. Very minor experience with Kotlin language and C/C++ language. Previous experience with a book renting service with the use of a database through Intelli-J’s IDE.

## Jaylen Cook - Role

Familiarity with Java and Kotlin programming languages. Experienced with Android Studios, IntelliJ, and Netbeans IDE. Some experience in 2D and 3D game design using Gamemaker Studio and Unity respectively. Experienced with GitHub.

## Benjamin Johnson - Role

Familiarity with Java and JavaFX, minor experience with other data-oriented, imperative, and object-oriented languages running on Linux, and previous work on a web-based application running on a server.

# **Dependencies, Limitations, and Risks**

## Dependencies

GitHub services may possibly fail, causing errors or corruptions in files. A lack of good practices may create risk. A free provided service may change their applications or policies. ChatGPT may provide incorrect guidance and cause a break or failure in code or services. Policies and licenses for services may change, causing an alteration with services. Google or Android Play services may experience an outage causing a loss in service.

## Limitations

Learning a new programming language, Kotlin. The group will also be using an IDE that they may not be entirely familiar with. The group is also using Android Studio for the IDE, and will be using ChatGPT for simple guidance and instruction, but for not code production.

## Other Risks

Risk may occur from work life balances, such as working a full time job and meeting deadlines. Acts of god such as a computer completely crashing resulting in data loss. The destruction of a device.

# **Timeline**

## Proposal Stage (8/26/25 - 9/13/25)

Complete the project proposal, and identify any current, or future risk. Identify the software, frameworks, IDE’s, and any operating systems that will be used for the creation of the project. The team will get all computers and laptops installed with the decided upon software the group has agreed to use. The group also plans to have scheduled group meetings, virtually and in person, to discuss these topics in further detail, till the end of the project to help facilitate communication and improve the quality of the project. The group also has to meet in and around others schedules.

## Training and Adjusting to Tools and IDEs (9/14/25 - 9/27/2025)

The team will get familiar with their software and further their experience in Kotlin, Android Studio, SQLite, Figma, GitHub, and any other necessary tools.

## App/UI Design (9/28/25 - 10/11/25)

Begin the base creation of the application and designing the user interface. Determine the main UI and which buttons link to which UI page.

10/5/25:

* Start finalizing Main UI, start focusing on sub pages and other smaller UI elements

## App Creation/Code Implementation (10/12/25 - 11/8/25)

10/12/25 - 10/19/25: Planning

* Determine datatypes and data structures.
* Determine which features need to be implemented first.
* Determine which member implements what feature.

10/20/25 - 11/8/25: Implementation

## Testing and Debugging (11/9/25 - 11/22/25)

Test all implemented features and fix as many bugs as possible.