

Product Requirements Document: Greasing the Groove (Android)

Document Version: 1.6

Date: October 12, 2025

Core Ethos: "Long-term consistency beats short-term intensity."

1. Overview and Context

1.1 Product Goal

The primary goal of **Greasing the Groove** is to help busy individuals establish and maintain a consistent daily strength habit by focusing on completing small, manageable goals (daily sets or daily hold durations) spread throughout the day, following the Greasing the Groove (GtG) principle. The app should minimize friction in logging and maximize positive reinforcement for consistency (streaks).

1.2 Target User Persona

Name: Alex

Age: 35-45

Profession: Manager/Professional (Marketing, Tech, Finance)

Constraint: Time-poor, often has unpredictable schedules.

Current State: Finds traditional 60-minute gym sessions intimidating or impossible to stick to.

Need: Wants to feel successful with a few minutes of effort several times a day (e.g., 5 sets of pull-ups, 10 minutes total of plank holds) without logging complex "workouts." The process must be seamless and quick.

1.3 Problem Statement

Traditional fitness apps focus on *sessions* (e.g., "Today's workout: Chest and Tris") which requires a large, dedicated chunk of time, leading to high failure rates and eventual habit abandonment for busy users.

1.4 Solution (Greasing the Groove)

Greasing the Groove abstracts away the "workout session" and replaces it with **Daily Set Goals** (for repetition-based movements) or **Daily Duration Goals** (for isometric holds). Users define a simple goal, and the app's entire experience is designed around quickly logging these sets/durations as they occur throughout the day, and celebrating the achievement of the daily goal and the resulting consistency streak.

2. Key Performance Indicators (KPIs) and Success

Metrics

Success for Greasing the Groove is measured by habit formation and consistency, not intensity or volume.

Metric ID	Metric Name	Definition	Priority
KPI-1	Active Streak Length (Primary)	The longest consecutive number of days a user meets their set goals for at least one exercise.	P1 (Critical)
KPI-2	Daily Goal Completion Rate	Percentage of users who complete 100% of their defined daily sets/duration goals.	P2 (High)
KPI-3	Sets/Duration Logged Per Day	Average number of sets or total duration (normalized) logged across all active users.	P3 (Medium)
KPI-4	Notification Effectiveness	Percentage of set completions within 15 minutes of a scheduled reminder.	P2 (High)

3. Core Feature Requirements

3.1 Goal Definition and Setup (P1)

- **FR-3.1.1: Exercise Library:** Allow users to select common bodyweight or simple equipment exercises *and* any custom exercises they have defined.
- **FR-3.1.2: Daily Goal Configuration:** Users **must** be able to define an active daily goal. This goal is saved in the activeGoals collection and references a defined exercise.
 - **Type 1: Sets (Count-Based):** Set a specific *total number of sets* (e.g., "Goal: 10

sets" of Push-ups).

- **Type 2: Duration (Time-Based/Isometric):** Set a specific *total duration* in minutes/seconds (e.g., "Goal: 10 minutes total" of Planks).
- **FR-3.1.3: Resting Goal:** The app must default to a 24-hour goal period (00:00 to 23:59).
- **FR-3.1.4: Custom Exercise Creation:** Users must be able to define a new exercise by providing a **Name** and immediately selecting its **Exercise Mode** ('REPS' or 'DURATION'). Once created, the exercise is available in the library and ready for goal configuration.

3.2 Quick Set Logging (P1)

This is the most critical feature. The UI must adapt based on the exercise's goal type.

- **FR-3.2.1: Primary Logging Action (Reps Mode):** For Reps Mode exercises, the main screen must display a large, immediately tappable button that instantly logs **one set** for that exercise, using the **last recorded settings** as defaults.
- **FR-3.2.2: Primary Logging Action (Duration Mode):** For Duration Mode exercises, the button must initiate the **Isometric Timer UI** (see FR-3.2.4).
- **FR-3.2.3: Detail Input (Reps, Weight, Notes - Optional):** Upon logging a set (either Count or Duration based), the app should briefly show a small, dismissible card/dialog allowing the user to optionally input:
 - The number of reps (Reps Mode only).
 - The amount of weight added/resistance used (applies to both modes).
 - A quick note.This input must be easily skipped. The default action is logging the set/duration count only, using the last recorded settings for pre-fill.
- **FR-3.2.4: Isometric Timer UI:**
 - When activated, a simple, large, full-screen timer interface must appear with "START" and "STOP" actions.
 - The timer must accurately track the held duration.
 - Tapping "STOP" automatically logs the duration to the `dailySetLogs` collection and returns the user to the dashboard. The optional detail card (FR-3.2.3) will appear briefly.
- **FR-3.2.5: Immediate Feedback:** Provide haptic feedback and a celebratory visual cue upon successful set/duration completion, showing the updated progress towards the daily goal.

3.3 Consistency and Progress Tracking (P2)

- **FR-3.3.1: Streak Visualization:** The main dashboard must prominently display the current active streak length (KPI-1) as the primary measure of success.
- **FR-3.3.2: Homepage Calendar View:** The homepage must feature a prominent calendar view where each day is marked to indicate daily consistency: Green (Goal Met), Yellow (Goal Partially Met), or Red/Grey (Goal Failed/Skipped). The calendar should be interactive.
- **FR-3.3.3: Historical Set/Duration Tracking:** Allow users to view total counts or duration

logged for a given exercise over the last week, month, and all time.

- **FR-3.3.4: Customizable Progress Graphs:** Users must be able to add, remove, and customize graph widgets on the homepage.
 - Each graph must track a **single metric** for a **single exercise** over a chosen time period.
 - Plottable metrics include: **Max Weight Added per Day**, **Average Reps per Set**, and **Total Daily Duration** (for time-based exercises).

3.4 Habit Reinforcement and Reminders (P2)

Since GtG relies on spreading sets out, reminders are essential.

- **FR-3.4.1: Customizable Reminders:** Allow users to define multiple daily reminder times for each exercise (e.g., "Push-up reminder at 9:00 AM, 1:00 PM, and 5:00 PM").
- **FR-3.4.2: Contextual Notifications (Android Feature):** Notifications should include a deep link or quick action button to log a set directly from the notification shade, fulfilling FR-3.2.1.
- **FR-3.4.3: Goal Completion Notification:** Send a final celebratory notification when the user hits 100% of their daily sets goal for a specific exercise.

4. Technical and Data Requirements

4.1 Platform and Development

- **Platform:** Android (Primary focus).
- **Database:** Utilize Firestore for user data storage and real-time syncing.
- **Auth:** Mandatory use of the provided Canvas authentication global variables (`__app_id`, `__firebase_config`, `__initial_auth_token`) for user identification and data isolation.

4.2 Core Data Model Concept (Firestore)

The data model now cleanly separates exercise definitions (static) from active goals (dynamic).

Collection 1: Exercise Definitions (Static Metadata and Last Settings)

/artifacts/{appId}/users/{userId}/exercises

Field Name	Type	Description
id	STRING	Unique ID for the exercise (e.g., 'pushups', 'planks').
name	STRING	Display name of the exercise (e.g., 'Push-ups', 'Plank').

exerciseMode	STRING	Must be 'REPS' or 'DURATION'. Defines the primary way the exercise is performed.
isCustom	BOOLEAN	true if this exercise was created by the user (FR-3.1.4).
lastReps	NUMBER (Optional)	Last recorded reps (default pre-fill for Reps Mode logging).
lastWeightAdded	NUMBER (Optional)	Last recorded weight added (default pre-fill for logging).
lastDurationSeconds	NUMBER (Optional)	Last recorded duration (default pre-fill for Duration Mode logging).
dateCreated	TIMESTAMP	When this exercise definition was first added.

Collection 2: Active Daily Goals (The Target)

/artifacts/{appId}/users/{userId}/activeGoals

(This collection should typically hold only one document per active exercise.)

Field Name	Type	Description
exerciseld	STRING	Foreign key referencing the exercises ID.
dailyTargetSets	NUMBER (Optional)	Target number if the linked exercise's mode is 'REPS'.
dailyTargetDurationSeconds	NUMBER (Optional)	Target duration if the linked exercise's mode is 'DURATION'.
dateSet	TIMESTAMP	When this goal was last configured/activated.

Collection 3: Daily Logged Sets (The Progress)

/artifacts/{appId}/users/{userId}/dailySetLogs (No changes to fields)

Field Name	Type	Description
id	STRING	Auto-generated log ID.
exerciseld	STRING	Foreign key referencing the exercises ID.
date	STRING	The date the set was logged (YYYY-MM-DD format for easy querying/grouping).
timestamp	TIMESTAMP	Exact time of logging.
weightAdded	NUMBER (Optional)	Amount of weight/resistance used for this set/hold.
reps	NUMBER (Optional)	Number of reps completed in this set (used if exercise mode is 'REPS').
durationSeconds	NUMBER (Optional)	Duration held in seconds (used if exercise mode is 'DURATION').
notes	STRING (Optional)	User notes for the set.

Collection 4: Graph Configurations

/artifacts/{appId}/users/{userId}/graphConfigs (No changes to fields)

Field Name	Type	Description
id	STRING	Unique ID for the saved graph configuration.
exerciseld	STRING	ID of the exercise being plotted (references the

		exercises collection).
metricType	STRING	Metric to plot: 'MAX_WEIGHT', 'AVG_REPS', or 'TOTAL_DURATION'.
period	STRING	Time range: 'LAST_30_DAYS', 'LAST_90_DAYS', or 'ALL_TIME'.
order	NUMBER	Display order on the homepage dashboard.

4.3 LLM Development Context

Future development prompts will leverage this PRD by referencing sections:

1. **Code Prompts (Database):** Must implement the four-collection data model described in 4.2. **Logging a set (Collection 3) must trigger an update to the last... fields in the corresponding exercises document (Collection 1).**
2. **Code Prompts (Logic):** Goal tracking must look up the target from the **activeGoals collection** and compare it against the aggregated logs from the **dailySetLogs collection**.

END OF DOCUMENT