Push-Up Tracker & 10-Minute Challenge

Product Requirements Document

---

1. Overview

The Push-Up Tracker & 10-Minute Challenge application is designed to help users accurately track their push-up workouts in real-time. The system provides immediate feedback on correct form and offers a timed challenge mode, encouraging users to push their limits and stay motivated.

---

2. Objectives

1. **Provide Real-Time Feedback**
   Detect and analyze user movements, showing whether they are in the correct form and counting repetitions accurately.

2. **Engage Users with a Time-Based Challenge**
   Offer a 10-minute mode for users to measure performance within a defined timeframe, fostering competition and goal-setting.

3. **Ensure Ease of Use**
   The application should be intuitive and require minimal setup, such as simply granting access to the user's camera (if applicable).

---

3. Scope

- **In Scope**
    - Real-time detection of user movements (arms, elbows, wrists) and push-up posture.

    - Rep counting based on confirmed transitions between “up” and “down” positions.
    - A timed challenge mode, which stops the session automatically upon timer expiration.
    - Visual feedback displayed to the user, including posture status (“Not set”, “Set”) and motion status (“Up”, “Down”, or “Hold”).
    - Summary screen to display total push-ups completed.
- **Out of Scope**
    - Comprehensive fitness analytics (e.g., storing user metrics over time, advanced analytics).
    - Integration with hardware or wearables outside a standard camera/webcam.
    - Multi-user or collaborative workout features.

---

4. Target Audience and Stakeholders

- **Primary Users:**
    - Fitness enthusiasts who want immediate feedback on their form and progress during push-ups.
    - Casual home exercisers looking for a straightforward way to count push-ups and maintain motivation.
- **Secondary Stakeholders:**
    - Product managers overseeing fitness and health applications.
    - Development teams working on AI-driven workout-tracking solutions.
    - Potential partners or fitness influencers interested in promoting a branded version.

---

5. Core Features and Requirements

5.1 Real-Time Pose Recognition

1. **Detection and Analysis**
    - Capture the user's movement through a camera feed (or equivalent motion-sensing technology).
    - Track key body points (e.g., shoulders, elbows, wrists) to determine posture.
2. **Rep Counting**
    - Confirm a valid push-up repetition when transitioning from an "up" (arms extended) to a "down" (arms bent) position and back.
    - Maintain a counter of total push-ups.
    - Provide on-screen feedback (e.g., "Up" or "Down") to confirm movement.
3. **Visual Indicators**
    - Display a live feed of the user's movement, overlayed with markers indicating recognized body points (optional, depending on design choices).
    - Mark statuses (e.g., "Not set" vs. "Set" position) in real time.

5.2 10-Minute Challenge

1. **Timer**
    - Show a countdown timer (10 minutes by default) visible on the workout screen.
    - End the session automatically when time runs out.
2. **Challenge Summary**
    - Present total push-ups completed during the 10-minute period.
    - Optionally allow the user to restart or proceed to a new session.

5.3 User Interface

1. **Landing/Start Screen**
    - Offer two modes: Standard Workout and 10-Minute Challenge.
    - Provide brief instructions or guidance on how to position the camera and themselves.
2. **Workout Screen**
    - Display the live feed (or user interface for motion feedback) and the push-up counter.
    - Indicate motion status (e.g., “Up” in green, “Down” in red) and position status (e.g., “Set” in green, “Not set” in red).
    - If in challenge mode, display the remaining time.
    - Provide a button to end the session at any time.
3. **End/Results Screen**
    - Present the final push-up count and, if challenge mode was used, show the time remaining or confirm that time ended.
    - Let the user start a new workout or close the application.

---

6. Use Cases

6.1 Start a Standard Workout

- **Actor:** User
- **Goal:** Begin a push-up session without time constraints.
- **Process Flow:**
    1. User launches the application.
    2. User selects “Standard Workout.”

3. The system accesses the user's camera (or motion sensor).
4. The workout screen displays, and rep counting begins.

6.2 Start the 10-Minute Challenge

- **Actor:** User
- **Goal:** Attempt as many push-ups as possible within 10 minutes.
- **Process Flow:**
    1. User selects "10-Minute Challenge" from the start screen.
    2. The system displays a timer and tracks push-ups in the same manner as the standard session.
    3. Upon timer expiration (or manual ending), the system shows the session summary.

6.3 Count a Valid Push-Up

- **Actor:** User / System
- **Goal:** Accurately detect and confirm a correct push-up repetition.
- **Process Flow:**
    1. User transitions from up position (arms extended) to down position (arms bent).
    2. The system confirms the movement over a short threshold (e.g., multiple frames) to avoid false positives.
    3. When transitioning back to "up," the counter increments by one rep.
    4. The system updates the on-screen counter and provides visual feedback.

6.4 End Workout and View Results

- **Actor:** User

- **Goal:** Stop the current session and view push-up count.
- **Process Flow:**
    1. User presses “End Workout,” or the session concludes when the challenge timer expires.
    2. The system stops tracking and shows the workout summary (total push-ups).
    3. The user can choose to start a new workout session or exit the application.

---

7. Non-Functional Requirements

1. **Usability**
    - Intuitive design enabling users to start and end workouts easily.
    - Clear on-screen instructions and status indicators.
2. **Performance**
    - Real-time detection with minimal latency.
    - Efficient handling of camera input or motion data.
3. **Device Compatibility**
    - Support various devices, if possible, ensuring responsive design.
    - Allow fallback or alternative instructions on devices lacking a standard camera.
4. **Security and Privacy**
    - Inform users about camera usage.
    - Process video or motion data locally as much as possible, avoiding unnecessary data transmission or storage.
5. **Scalability and Extensibility**

- Provide a modular design that can be extended for future exercise types or challenges.
- No requirement on the specific technology stack or programming language, allowing flexible implementation.

---

8. Risks and Assumptions

- **Risks**
  - Performance may vary across different devices and operating systems.
  - Incorrect lighting or camera angles could reduce pose detection accuracy.
- **Assumptions**
  - Users have a device with a camera (or equivalent sensor) and reasonable processing power.
  - Users are physically able to perform push-ups in a clear area where they can be detected.

---

9. Success Criteria

- **High user engagement:** Users find the app simple to use, start sessions regularly, and complete the challenge.
- **Accurate counting:** Users report that the push-up counts match their actual reps.
- **Positive feedback:** Users feel encouraged by real-time motion feedback and effectively complete workouts.

---

10. Future Enhancements

- **Additional Exercises:** Extending the detection logic to track squats, lunges, or other movements.
- **Progress Tracking:** Ability to log historical data for users to track long-term trends.
- **Gamification:** Leaderboards, badges, or challenges with friends for higher engagement.