

Software Requirements Specification (SRS) for a Workout Tracker Application

Team:

NM-LICET-IT-GROUP2

Naveen S (au311120205042)

Thaswin Meshak A (au311120205058)

Bhuvanesh Raam CK (au311120205017)

Aswin K (au311120205301)

Introduction:

Purpose: Elaborate on the need for the workout tracker app. Discuss the current challenges faced by fitness enthusiasts in tracking their workouts and setting goals, emphasizing the app's role in addressing these challenges.

Scope: Define the boundaries of the app by detailing what functionalities it will include (e.g., workout logging, progress visualization, goal setting) and what it won't cover (e.g., personalized diet plans).

References: Provide links or references to industry standards, fitness guidelines, or similar apps for comparative analysis.

Functional Requirements:

User Management: Specify user roles and their permissions (admin, regular user, trainer) along with the registration, login, and password reset processes.

Workout Tracking: Describe how users will log their workouts. Include details about exercise selection, adding duration, intensity levels, repetitions, sets, equipment used, and optional notes.

Progress Monitoring: Explain how users can visualize their progress. This might include generating charts, graphs, or comparative statistics showcasing improvements over time.

Goal Setting: Detail the process of setting, modifying, and tracking fitness goals, allowing users to specify objectives and deadlines.

Notifications and Reminders: Describe how the app will send reminders for upcoming workouts or milestone achievements through in-app notifications or emails.

Non-Functional Requirements:

Performance: Specify response time requirements for different actions within the app to ensure a smooth user experience even during peak usage.

Usability: Define interface guidelines, emphasizing user-friendly design, intuitive navigation, and accessibility features.

Security: Detail measures like encrypted storage, secure authentication, and data protection to safeguard user information.

Compatibility: List supported platforms (iOS, Android) and device specifications to ensure the app functions seamlessly across various devices.

Maintainability: Describe how the app will handle updates, scalability, and future enhancements without disrupting user experience.

User Interface Requirements:

Wireframes or Mockups: Include detailed sketches or digital representations of the user interface design, highlighting layout, colors, typography, and button/icon placements.

UI Design Guidelines: Provide style guides defining design principles, visual elements, and interactive components.

System Requirements:

Hardware Requirements: Specify server configurations, storage capacities, and hardware components needed to host and run the application.

Software Dependencies: List required software, frameworks, libraries, and APIs needed for development, deployment, and third-party integrations.

Third-Party Integrations: Outline integrations with external services like fitness device APIs (e.g., Fitbit, Apple Health).

Constraints and Assumptions:

Constraints: Detail limitations such as budget constraints, time restrictions, technological limitations, or regulatory compliance requirements.

Assumptions: Document assumptions made during requirement gathering, outlining potential risks and their impact if these assumptions prove incorrect.

Use Cases/Scenarios:

User Stories: Detail various personas (users, trainers, administrators) and their interactions with the app. For example: User Persona: "Fitness Enthusiast" Scenario: The user logs in, selects a workout plan, records exercises performed with sets and reps, tracks progress, and receives achievement notifications.

Flow Diagram: Visualize the user's journey, depicting steps like login, workout logging, progress visualization, and goal setting.

Edge Cases: Include scenarios that fall outside typical interactions to ensure the application handles exceptional situations effectively. For instance: Edge Case: "Network Failure During Workout Logging" Scenario: The user loses internet connectivity while logging a workout session. Expected Behavior: The app stores workout data locally and syncs it when the network connection is restored.

Exception Handling: Describe how the application should respond in case of errors or unexpected user actions.

Glossary:

Definitions: Compile a comprehensive list of terms, acronyms, and technical jargon used throughout the document and their respective definitions. For instance: HIIT (High-Intensity Interval Training): A workout method involving short bursts of intense exercise followed by brief recovery periods. API (Application Programming Interface): A set of rules allowing different software applications to communicate with each other. OAuth (Open Authorization): An open standard for authorization allowing users to share their private resources stored on one site with another site without having to share credentials.

Explanation of Concepts: Provide detailed explanations or descriptions for complex concepts specific to fitness tracking or technological aspects pertinent to the application's development. This ensures a common understanding among stakeholders who may have varying levels of expertise in the domain.

Appendix:

Research Findings: Include research papers, market analysis reports, or surveys related to fitness trends, user preferences, or competitor analysis that influenced the app's design or feature set.

User Feedback: Attach summaries or transcripts of user interviews, surveys, or feedback sessions conducted during the requirement gathering phase. Highlight insights and preferences expressed by potential users regarding workout tracking and goal setting.

Supplementary Documents: Provide additional documents such as architectural diagrams, data flow diagrams, or external API documentation relevant to the application's development and integration.

Legal and Compliance Documentation: If applicable, include documents related to data privacy laws, compliance standards (e.g., GDPR), and terms of service that the application must adhere to.