Software Requirements Specification

for

Fitness Web App

Version 1.0 approved

Prepared by :

Hanesh Iyer

Jayesh Nagnare

Kamlesh Kaparvena

Vidyalankar Institute of Technology

24 Aug , 2023

Table of Contents

1. Introduction

1.1 Purpose

The purpose of this Fitness Web App Software Requirements Specification (SRS) document is to outline the functional and non-functional requirements for the development of a fitness-oriented web application. This document serves as a guideline for the development team, stakeholders, and users to understand the features, behavior, and constraints of the system.

1.2 Document Conventions

Term	Definition
DB	Database
DDB	Distributed Database
ER	Entity Relationship
Gym Client	Person who is using the services of the gym and the mobile application.
Fitness IT manager	System administrator who manages and maintains The fitness organization’s technology.
Android	A mobile devices operating system designed by Google Inc.
IOS	An Operating System created and developed by Apple Inc.

1.3 Intended Audience and Reading Suggestions

This SRS document is intended for programming developers, product testers, project managers, marketing staff; and our stakeholders may review this as a guideline into understanding the requirements and maintain the main goal of the application. By reading this document, the reader will get the general understanding of the product including the application's functional and non-functional requirements.

1.4 Product Scope

The Fitness App is a Web application that will run on the main mobile platform (iOS and Android devices). The application will provide a way for the gym users to manage their active lifestyles such as to book an appointment with a trainer, and to create a workout plan based on the customer's goals (either by getting toned, lose weight, maintaining physical healthy body, etc.). Fitness App can also track your nutrition plan based on the customer's needs. By doing this, gym users will be able to plan and organize their desired physical goal freely without hassle.

1.5 References

IEEE.*IEEE* Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications. IEEE Computer Society, 1998.

Health and Fitness Apps Promotional Methods: Catching the Wave of New Year Resolutions. (2018, August 29). Retrieved January 14, 2019, from http://www.businessofapps.com/news/health-and-fitness-apps-promotional-methods-catching- the-wave-of-new-year-resolutions/

Elkaim, Y. (2018, October 30). How to Find Your Target Audience (For Health & Fitness Experts). Retrieved January 14, 2019, from https://healthpreneurgroup.com/how-to-find- your-target-audience/

1.6 Product Perspective

The Fitness Web App is designed to provide users with a comprehensive fitness management platform built using React.js for the front-end, MongoDB for the database, Express.js for the server, and Node.js for the runtime environment. It also integrates with the Rapid API to access exercise-related data. The product perspective encompasses the following aspects:

System Context:

The Fitness Web App operates as a standalone web application.
It interacts with the user's web browser for rendering the interface and handling user interactions.
It communicates with the MongoDB database for data storage and retrieval.
Express.js handles HTTP requests and responses between the client and server.
Node.js provides the runtime environment for the server-side components.

Interfaces:

External interfaces include the RESTful APIs provided by Rapid API for accessing exercise-related data.
The app uses MongoDB as its database interface for data storage.
It may have third-party integrations or APIs for features like social media sharing or payment processing.

Dependencies:

React.js relies on various libraries and packages, including state management tools like Redux or MobX, routing libraries like React Router, and UI component libraries like Material-UI or Ant Design.
Express.js and Node.js have dependencies for handling server-side tasks, including authentication middleware, database drivers, and security packages.
MongoDB serves as the primary data storage solution, and its setup and maintenance are essential dependencies.

Constraints:

Compatibility with various web browsers, including Chrome, Firefox, Safari, and Edge, is a key constraint to ensure a consistent user experience.
Compliance with the Rapid API's terms of use and rate limits is critical to maintain access to exercise-related data.
The app's performance and scalability depend on the hosting infrastructure and server resources available.

Assumptions:

The app assumes that users have access to a modern web browser and an internet connection.
Users are expected to enable JavaScript in their browsers for full functionality.
The server-side infrastructure assumes adequate resources for handling expected traffic and data storage.

Compatibility:

The Fitness Web App is designed to be compatible with a wide range of browsers, including the latest versions of Chrome, Firefox, Safari, and Edge. It is also expected to work seamlessly on various operating systems, such as Windows, macOS, Linux, and mobile platforms.

1.7 Product Functions

1.User Registration and Profile Management

1.1. **Register Account**: Users can create a new account by providing their email, username, and password or by using their social media accounts.

1.2**. Login**: Registered users can log in using their credentials.

1.3. **Password Recovery:** Users can request a password reset link if they forget their password.

1.4**. Update Profile**: Users can edit and update their profile information, including their name, profile picture, and contact details.

1.5. **Admin User Management**: Administrators can manage user accounts, including the ability to suspend, delete, or modify user profiles.

2. Personalized User Profiles

2.1. **Customize Profile**: Users can personalize their profiles by setting fitness goals, adding workout preferences, and defining privacy settings.

2.2**. Progress Tracking**: Users can log and track their fitness progress, including weight, body measurements, and workout achievements.

2.3**. Achievements and Badges**: Users can earn achievements and badges for reaching specific milestones or completing challenges.

3. Workout Tracking

3.1. **Create Workout**: Users can create custom workout routines by selecting exercises from the exercise library.

3.2. **Log Exercises**: Users can log individual exercises, recording details such as sets, reps, weights, and rest times.

3.3. **View Workout History**: Users can review their workout history to track performance improvements over time.

3.4**. Performance Metrics**: Users can view performance metrics for each exercise, including personal records and progress graphs.

3.1 User Classes and Characteristics

Regular Users

Characteristics:

Fitness Enthusiasts: Regular users are individuals passionate about fitness, including gym-goers, athletes, or those aiming for a healthier lifestyle.

Varied Fitness Levels: They may have different fitness levels, ranging from beginners to advanced athletes.

Diverse Goals: Regular users may have diverse fitness goals, such as weight loss, muscle gain, endurance improvement, or overall health.

Tech-Savvy: They are comfortable using web and mobile applications for tracking workouts, nutrition, and progress.

Social Interaction: They engage with friends and other fitness enthusiasts, sharing achievements and seeking motivation.

3.2 Operating Environment

Supported Browsers: The app is compatible with modern web browsers such as Chrome, Firefox, Safari, and Edge.

Devices: It runs on various devices, including desktops, laptops, tablets, and smartphones.

Internet Connectivity: Users need an internet connection to access and use the app.

JavaScript: Users are required to enable JavaScript in their browsers for the app's full functionality.

3.3 System Feature

3.3.1 Description and Priority.

1. User Registration and Authentication.

Description: Allow users to create accounts, providing necessary personal information. Enable secure authentication to ensure user data privacy.

Priority: High

2. User Profile Management

Description: Allow users to create and manage their profiles, including details like age, gender, weight, height, and fitness goals.

Priority: Medium

3. Workout Plans

Description: Provide users with workout planning tools, allowing them to create, customize, and track workout routines tailored to their goals and preferences.

Priority: High

4. Progress Tracking

Description: Enable users to log their workout performances and track their progress over time. Display visualizations such as graphs and charts to illustrate improvements.

Priority: High

5. Workout Challenges

Description: Introduce workout challenges with predefined routines and goals to motivate users. Provide rewards or recognition for completing challenges.

Priority: Medium

6. Calorie Calculator

Description: Integrate a calorie calculator that helps users estimate their daily caloric needs based on their profile information and fitness goals. Enable users to track calories burned during workouts.

Priority: Medium

4.1.2 Functional Requirements

1. User Registration and Authentication

1.1. The system shall provide a user registration form to capture user information, including name, email, password, age, and gender.

1.2. The system shall validate the uniqueness of email addresses during the registration process.

1.3. Users shall be able to log into the system using their registered email and password.

1.4. The system shall implement password hashing and salting for secure authentication.

2. User Profile Management

2.1. Users shall be able to create, edit, and update their profiles, including personal information such as age, gender, weight, and height.

2.2. Users shall have the option to upload a profile picture to personalize their profiles.

2.3. The system shall allow users to set and modify their fitness goals, such as weight loss, muscle gain, or general fitness improvement.

3. Progress Tracking

5.1. The system shall enable users to log their workout performance, including sets, repetitions, and weights lifted.

5.2. Users shall be able to view their workout history and progress over time through graphical visualizations.

5.3. The system shall provide analytics on exercise performance, showcasing improvements or areas that need attention.

4. Workout Plans

14.1. The system shall provide pre-designed workout plans for various fitness goals, such as weight loss, muscle gain, and endurance improvement.

14.2. Each workout plan shall include a list of exercises, the order in which they should be performed, recommended sets and repetitions, and progression guidelines.

14.3. Users shall be able to browse and select workout plans based on their fitness goals and experience level.

7. Calorie Calculator

7.1. The system shall provide a calorie calculator feature to estimate users' daily caloric needs based on their profiles and fitness goals.

7.2. Users shall be able to adjust their activity levels to receive accurate caloric intake recommendations.

7.3. The system shall allow users to track calories burned during workouts and deduct them from their daily caloric intake.

9. Workout Challenges

9.1. The system shall present users with predefined workout challenges with specific goals and timeframes.

9.2. Users shall receive notifications and reminders related to active challenges and their progress.

9.3. Upon completing challenges, users shall receive virtual rewards or badges.

4. Other Nonfunctional Requirements

4.1 Performance Requirements

Response Time: The system shall provide a response time of no more than 2 seconds for any user interaction.

Load Time: The web pages should load within 3 seconds on a standard broadband connection.

Concurrency: The system should support at least 500 concurrent users without significant degradation in performance.

Database Performance: Database queries shall execute within 300 milliseconds on average.

4.2 Safety Requirements

Data Encryption: User data, especially passwords, should be stored using strong encryption techniques.

User Authentication: The system should support secure user authentication and authorization mechanisms, ensuring that users can only access their own data.

Secure Communication: All communication between the client and server should be encrypted using HTTPS to prevent data interception.

Data Privacy: The system should comply with relevant data protection regulations, such as GDPR, ensuring user data privacy and allowing users to manage their data.

4.3 Usability

Intuitive User Interface: The user interface should be intuitive and easy to navigate for users with varying levels of technical expertise.

Consistency: UI elements and design should be consistent throughout the application to provide a cohesive user experience.

Accessibility: The app should adhere to accessibility guidelines (such as WCAG) to ensure that users with disabilities can use the app effectively.

4.3 Software Quality Attributes

Cross-Browser Compatibility: The app should be functional and visually appealing across popular web browsers (Chrome, Firefox, Safari, Edge).

Mobile Responsiveness: The app should be responsive and usable on various mobile devices and screen sizes.

Screen Readers: The app should be compatible with screen readers and other assistive technologies for visually impaired users.

5. Other Requirements

The Project require a database for storing data so we use MongoDB and Rapid Apis

Appendix A: Glossary

API: Rules for software communication.

Authentication: Confirming user identity.

Dashboard: Information summary.

Encryption: Data protection.

Exercise Library: Exercises database.

Front-end: User interface.

Goal Setting: Fitness objectives.

HTTPS: Secure data transfer.

Middleware: Software bridge.

Notification: User messages.

Progress Tracking: Fitness monitoring.

Responsive Design: Screen adaptability.

RESTful API: HTTP-based API.

User Profile: Personal page.

UX: User experience.

UI: Visual interface.

Workout Routine: Exercise plan.

Workout Tracking: Progress recording.

Server-Side: Server operations.

Client-Side: User device operations.

Appendix B: To Be Determined List

UI Mockups: Detailed app screen designs.

Database Structure: Final database schema.

Third-Party Integrations: Services to be added.

Notification Details: Types and content.

Legal Docs: Privacy policy, terms.

Content Moderation Guidelines: Community standards.

Performance Testing Plan: Load tests.

Security Measures: Authentication, encryption.

Scalability Strategy: Handling more users.