Software Requirement Specification (SRS) for Workout Tracker Application

Table of Contents

---

1. Introduction

1.1 Purpose

The purpose of this document is to provide a comprehensive Software Requirement Specification (SRS) for the Workout Tracker Application. This application aims to help users plan, track, and manage their workout routines effectively. It will be developed using ReactJS for the frontend, HTML and CSS for the user interface, Node.js for the backend, and MongoDB as the database.

1.2 Scope
The Workout Tracker Application will enable users to:
- Create and manage workout routines.
- Record and track individual exercises.
- View statistics and progress over time.
- Set goals and receive notifications.
- Register and authenticate users.

1.3 Definitions, Acronyms, and Abbreviations
- SRS: Software Requirement Specification
- UI: User Interface
- UX: User Experience
- API: Application Programming Interface

1.4 References
- Design and development best practices for ReactJS, HTML, CSS, MongoDB, and Node.js.
- [MongoDB documentation](https://docs.mongodb.com/)
- [ReactJS documentation](https://reactjs.org/docs/getting-started.html)
- [Node.js documentation](https://nodejs.org/en/docs/)

2. Overall Description

2.1 Product Perspective
The Workout Tracker Application is a standalone system that interacts with users through a web-based interface. It relies on ReactJS for the frontend and communicates with the Node.js backend via API calls. MongoDB is used to store and manage user data.

2.2 Product Features
The core features of the application include:
- User registration and authentication.
- User profile management.
- Workout routine creation and management.
- Exercise tracking and logging.
- Progress statistics and data visualization.
- Goal setting and notifications.

2.3 User Classes and Characteristics
- Regular Users: Individuals who want to track and improve their workout routines.
- Admin Users: System administrators responsible for managing user accounts and monitoring the system's overall performance.

2.4 Operating Environment
The Workout Tracker Application will be accessible through web browsers on various devices, including desktop computers, tablets, and smartphones. It will support modern browsers such as Google Chrome, Mozilla Firefox, Safari, and Microsoft Edge.

2.5 Design and Implementation Constraints
- The frontend will be built using ReactJS with a responsive design to ensure usability on different screen sizes.
- The backend will be implemented using Node.js, and MongoDB will be used as the database.
- The application will use RESTful API endpoints for communication between the frontend and backend.

2.6 Assumptions and Dependencies
- The application assumes a stable internet connection for users to access and use the platform.
- Users are expected to have basic knowledge of using web-based applications.

3. Specific Requirements

3.1 Functional Requirements
1. User Authentication
   - Users can register for an account.
   - Users can log in and out.
   - Passwords should be securely hashed and stored.

2. User Profile
   - Users can update their profile information.
   - Users can upload a profile picture.

3. Workout Routines

- Users can create, edit, and delete workout routines.
- Routines should have a name, description, and list of exercises.

4. Exercises
- Users can add, edit, and delete exercises within a routine.
- Exercises should have a name, description, sets, reps, and weight.

5. Logging Workouts
- Users can log individual workouts, associating them with specific routines and exercises.
- Users can input data for sets, reps, and weight lifted.

6. Progress Tracking
- The application will calculate and display statistics on workout progress.
- Users can view charts and graphs of their progress over time.

7. Goal Setting
- Users can set fitness goals, such as weight loss or muscle gain.
- Users can receive notifications and reminders related to their goals.

8. Admin Functionality
- Admin users can manage user accounts.
- Admin users can monitor system performance.

3.2 Non-functional Requirements

1. Performance
- The system should respond to user interactions within 2 seconds.
- The database should handle concurrent requests efficiently.

2. Security
- User data should be securely stored and transmitted.
- Passwords must be hashed and salted before storage.
- User sessions should be managed securely.

3. Scalability
- The system should handle an increasing number of users and data without performance degradation.

4. Compatibility
   - The application should be compatible with modern web browsers.
   - The user interface should be responsive and adapt to various screen sizes.

4. System Models

4.1 Use Case Diagram
(Insert a use case diagram illustrating the interactions between users and the system.)

4.2 Activity Diagram
(Insert an activity diagram representing the workflow of a typical user within the system.)

4.3 Database Schema
(Provide a description of the database schema, including tables, fields, and relationships.)

5. User Interface Design

5.1 Wireframes
(Include wireframes for key user interface screens, such as login, registration, profile management, routine creation, and workout logging.)

5.2 User Experience (UX) Design
(Describe the overall user experience design, including color schemes, typography, and any design principles followed.)

6. Appendix
(Include any additional information, diagrams, or references relevant to the SRS.)

This Software Requirement Specification (SRS) provides a comprehensive overview of the Workout Tracker Application's requirements and features. It serves as a guide for the development team to create a robust and user-friendly application using ReactJS, HTML, CSS, MongoDB, and Node.js.

Certainly, let's elaborate on each of the key features of the Workout Tracker Application:

1. Create and Manage Workout Routines:
- Create Routines: Users can create personalized workout routines tailored to their fitness goals and preferences. They can specify details such as the routine's name, description, and the days of the week it should be followed.

- Edit Routines: Users have the ability to make changes to their existing routines. For example, they can add new exercises, remove exercises, or adjust the number of sets and reps for each exercise.

- Delete Routines:If a user no longer wishes to follow a particular routine, they can delete it from their profile. This allows for easy management and organization of workout plans.

2. Record and Track Individual Exercises:
- Exercise Logging: Users can record their workouts by logging individual exercises. For each exercise, they can input data such as the number of sets performed, the number of repetitions in each set, and the weight lifted during each set.

- Workout History: The application maintains a workout history for each user, making it easy to review past exercises and track progress. Users can see their previous performance and identify areas for improvement.

- Exercise Details: Users can access detailed information about each exercise, including exercise names, descriptions, and proper techniques, ensuring they perform exercises correctly and safely.

3. View Statistics and Progress Over Time:
- Progress Tracking:The application calculates and presents statistics related to the user's fitness journey. This includes data such as total weight lifted, workout frequency, and improvements over time.

- Graphical Representation: Progress data is often presented visually through charts and graphs, allowing users to see their achievements and trends at a glance. This visual representation enhances the user's understanding of their progress.

- Historical Comparisons: Users can compare their current performance to their past records, helping them set new goals and adjust their workout routines for better results.

4. Set Goals and Receive Notifications:
- Goal Setting:Users can define specific fitness goals they aim to achieve. These goals might include objectives like losing a certain amount of weight, increasing muscle mass, or running a certain distance within a specified time frame.

- Goal Monitoring: The application assists users in monitoring their progress toward their set goals. It tracks their achievements and displays how close they are to reaching their desired milestones.

- Notifications: To keep users motivated and on track, the application can send notifications and reminders. For example, it might remind users of their workout schedule, provide encouragement when they achieve milestones, or suggest adjustments to their routines based on their goals.

5. Register and Authenticate Users:
- User Registration: New users can sign up for an account by providing necessary information, such as their email address, username, and a secure password. This process ensures that only authorized individuals can access the application.

- Authentication: Registered users can securely log in to their accounts using their credentials. Authentication is essential for protecting user data and ensuring the privacy and security of their workout information.

- User Profiles: Each user has a personal profile where they can manage their account settings, update their personal details, and upload a profile picture to customize their experience.

These elaborations provide a comprehensive understanding of the core features of the Workout Tracker Application and how they empower users to effectively plan, monitor, and achieve their fitness goals while maintaining a personalized workout routine.