Rg Fitness Club

Software Requirements Specification INT252

Utkarsh Sharma

12209451

SECTION: K22CD

Prepared for Continuous Assessment 3

Autumn 2024

## Revision History

**Date Description Author Comments**

|  |  |  |  |
| --- | --- | --- | --- |
| 28/10/2024 | This **Fitness and Nutrition Planner** site is a personalized wellness platform designed to help users manage their fitness routines and nutrition plans. Users can access a dashboard ,navigate through workout tracking, nutrition planning. | Utkarsh Sharma | Everyone can have their personalized plans. |

## Table of Contents

**REVISION HISTORY II**

**1. INTRODUCTION....................................................................................…………………………………………**

1.1.PURPOSE **………………………………………… …………………………………………………………….**

1.2.SCOPE ………………………………………… ………………………………………………………………………………………….

* 1. Definitions, Acronyms, and Abbreviations…………………………………………………………………….
  2. REFERENCES **ERROR! BOOKMARK NOT DEFINED.**
  3. OVERVIEW **ERROR! BOOKMARK NOT DEFINED.**

1. **GENERAL DESCRIPTION** ERROR! BOOKMARK NOT DEFINED.
   1. PRODUCT PERSPECTIVE **ERROR! BOOKMARK NOT DEFINED.**
   2. PRODUCT FUNCTIONS **ERROR! BOOKMARK NOT DEFINED.**
   3. USER CHARACTERISTICS **ERROR! BOOKMARK NOT DEFINED.**
   4. GENERAL CONSTRAINTS **ERROR! BOOKMARK NOT DEFINED.**
   5. ASSUMPTIONS AND DEPENDENCIES **ERROR! BOOKMARK NOT DEFINED.**
2. **SPECIFIC REQUIREMENTS 4**
   1. EXTERNAL INTERFACE REQUIREMENTS 5
      1. *User Interfaces 5*
      2. *Hardware Interfaces 5*
      3. *Software Interfaces 5*
      4. *Communications Interfaces 5*
   2. FUNCTIONAL REQUIREMENTS 5
      1. *<Functional Requirement or Feature #1>* ***Error! Bookmark not defined.***
      2. *<Functional Requirement or Feature #2>* ***Error! Bookmark not defined.***
3. NON-FUNCTIONAL REQUIREMENTS 3
   1. *Performance 3*
   2. *Reliability 3*
   3. *Availability 3*
   4. *Security 3*
   5. *Maintainability 3*
   6. *Portability 3*

3.7 DESIGN CONSTRAINTS 3

3.9 OTHER REQUIREMENTS 3

1. **ANALYSIS MODELS 12**
   1. DATA FLOW DIAGRAMS (DFD) 4
2. **GITHUB LINK 5**
3. **DEPLOYED LINK 6**
4. **CLIENT APPROVAL PROOF 7**
5. **CLIENT LOCATION PROOF 8**
6. **TRANSACTION ID PROOF 9**

**A. APPENDICES...........................................................................................................................................................**

A.1.APPENDIX 1...........................................................................................................................................................

A.2.APPENDIX 2...........................................................................................................................................................

# Introduction

**This Software Requirements Specification (SRS) document outlines the requirements for a web-based application designed for hotel room bookings. The primary objective of this platform is to provide users with a seamless, user-friendly experience when searching for, comparing, and reserving hotel accommodations.**

# Purpose

**The Software Requirements Specification (SRS) document outlines the core functionalities, system requirements, and user interactions to guide the development and deployment of the Fitness and Nutrition Planner. This document serves as a resource for developers, stakeholders, and quality assurance teams, providing detailed requirements to ensure the application meets user expectations for reliability, usability, and scalability.**

# Scope

**The Fitness and Nutrition Planner is a web-based application designed to help users manage and achieve their fitness and dietary goals by providing tailored exercise plans and nutritional guidance. The primary features within the scope of this application include:**

1. **Dashboard Interface: A welcoming dashboard where users can specify their fitness goals, target areas of the body for workouts, and preferred types of training. Based on this input, the application will generate a personalized workout schedule.**
2. **Schedule Generation and PDF Export: A customized workout schedule based on user preferences, with the option to download the plan as a PDF, providing flexibility for offline access and easy reference.**

# Overview

**The Software Requirements Specification (SRS) is organized into the following sections:**

* **Overall Description: Provides context and background of the Fitness and Nutrition Planner project, including user needs and system objectives.**
* **Functional Requirements: Details the specific functionalities the system must support, such as workout planning, AI chat assistance, and schedule downloads.**
* **Non-functional Requirements: Addresses performance, usability, and other quality attributes for the application.**
* **System Architecture: Outlines the technical structure, backend and frontend frameworks, and databases to be used.**
* **Assumptions and Dependencies: Lists any assumptions made during requirements gathering and dependencies on external technologies or APIs.**

**This SRS aims to ensure a clear understanding of the project requirements, serving as a foundation for the successful development and deployment of the Fitness and Nutrition Planner application.**

# Definitions, Acronyms, and Abbreviations Definitions

* **Fitness and Nutrition Planner: A web application built using React.js that helps users customize workout routines, track progress, and receive guidance on fitness and nutrition topics.**
* **User: An individual who interacts with the application to create, customize, and download workout schedules.**
* **Administrator: An individual with enhanced permissions responsible for managing content and monitoring usage.**
* **Workout Schedule: A tailored workout routine that the user can generate based on their chosen preferences and download as a PDF.**

# Acronyms

* **SRS: Software Requirements Specification**
* **UI: User Interface**
* **UX: User Experience**
* **API: Application Programming Interface**
* **HTML: Hypertext Markup Language**
* **CSS: Cascading Style Sheets**
* **JS: JavaScript**

# Abbreviations

* **FR: Functional Requirements**
* **NFR: Non-Functional Requirements**
* **PDF: Portable Document Format**
* **HTTP: Hypertext Transfer Protocol**
* **HTTPS: Hypertext Transfer Protocol Secure**

*This section provides clarity on the terms and concepts relevant to the development of the Fitness and Nutrition planner website using React.js, ensuring a common understanding among all stakeholders.*

*This subsection of the SRS should list each of the factors that affect the requirements stated in the SRS. These factors are not design constraints on the software but are, rather, any changes to them that can affect the requirements in the SRS. For example, an assumption might be that a specific operating system will be available on the hardware designated for the software product. If, in fact, the operating system is not available, the SRS would then have to change accordingly.*

**Specific Requirements**

**Functional Requirements**

User Interaction and Dashboard Customization

* FR-1: Upon site entry, the system shall display a Dashboard where users can select preferences related to fitness goals, such as body part focus (e.g., arms, legs, core) or specific types of training (e.g., strength, cardio).
* FR-2: Users shall have the option to download their customized workout schedule as a PDF.

Workout and Nutrition Planning

* FR-3: The system shall provide a recommendation engine for users to generate a workout plan based on the chosen preferences and goals.
* FR-4: The system shall integrate an AI-powered chatbot that provides real-time answers on fitness and nutrition topics.

PDF Generation and Download

* FR-5: The system shall allow users to download workout plans as PDF files for offline access.

AI Assistance

* FR-6: Users shall be able to ask questions related to fitness and nutrition in a conversational AI chat interface, which provides responses based on pre-trained models and fitness/nutrition knowledge.

Non-Functional Requirements Performance

* NFR-1: The system shall load the Dashboard page within 2 seconds under standard network conditions.
* NFR-2: The system shall support a minimum of 500 concurrent users with no noticeable performance degradation.

Security

* NFR-3: All data, including user preferences and workout schedules, shall be encrypted during transmission using SSL/TLS to ensure secure data handling.
* NFR-4: While direct login functionality is not required, the system will implement secure data storage practices, including encrypted storage for workout preferences.

Usability

* NFR-5: The user interface shall be designed for intuitive navigation, ensuring ease of access to fitness plans, AI chat, and downloadable content.
* NFR-6: The system shall be fully responsive, providing a seamless experience across desktop, tablet, and mobile devices.

Scalability

* NFR-7: The architecture shall be designed to scale efficiently, accommodating increased user traffic and potential feature expansions, such as diet planning or additional workout options.

Compatibility

* NFR-8: The system shall support the latest versions of major web browsers, including Chrome, Firefox, Safari, and Edge, to ensure compatibility and ease of use across platforms.

System Architecture

The system will be developed using React.js for the front end to create a responsive user interface and PHP for backend processes, such as workout schedule generation and AI chat responses. Data will be stored in MongoDB for managing user preferences and AI interactions. The application will be hosted on a scalable cloud platform (e.g., AWS or Azure) to ensure high availability and easy scalability.

External Interface Requirements User Interfaces

* The Fitness and Nutrition Planner will have a visually appealing and user-friendly Dashboard with options to select workout focus and download plans. It will also feature an interactive AI chat interface for fitness and nutrition questions.

Hardware Interfaces

* The application will be accessible on various devices, including desktops, tablets, and smartphones, ensuring broad accessibility.

Software Interfaces

* The system will communicate with backend APIs, utilizing RESTful services to retrieve workout plans, AI responses, and PDF generation services. MongoDB will be used for data management, storing user preferences, and AI training data.

Communications Interfaces

* All communication between the front end and back end will use secure protocols (HTTPS) to ensure user data security.

Overview

The Fitness and Nutrition Planner is designed to provide users with a personalized experience for creating workout schedules based on individual fitness goals and preferences. It also includes an AI-powered chat assistant for questions on fitness and nutrition, accessible directly from the Dashboard.

This SRS document is structured to provide clarity on the project’s requirements, guiding the development process to achieve an intuitive, secure, and scalable application that meets the needs of users seeking tailored workout and nutrition advice.

Overall Goal: These non-functional requirements will ensure a robust, secure, and user-friendly platform capable of supporting current and future demand in fitness and nutrition planning.

Performance Requirements

* Page Load Time: The dashboard and other pages shall load within 2 seconds under standard network conditions.
* Concurrent User Support: The system shall support a minimum of 500 concurrent users without degradation.
* Response Time: User actions like initiating workout/nutrition plans and retrieving AI-generated recommendations shall receive responses within 1 second under normal load.
* Data Retrieval: Workout and nutrition plans should be retrieved and displayed within 2 seconds.
* Scalability: The architecture shall support horizontal scaling to accommodate growing user demand.
* Optimization: Performance enhancements, like code splitting and lazy loading, shall improve load times, particularly for AI and multimedia features.
* Monitoring: Performance monitoring tools shall be integrated to track metrics and identify bottlenecks.

Reliability Requirements

* Uptime: Maintain a 99.9% uptime with minimal maintenance downtime.
* Data Integrity: Validation checks shall ensure data accuracy and prevent corruption, particularly for user-generated workout and nutrition data.
* Error Handling: Provide clear error messages and recovery options if data retrieval or saving fails.
* Backup and Recovery: Regular, automated database backups shall ensure recovery within 1 hour of failure.
* Transaction Reliability: All entries (e.g., workout logs) should only confirm success once completed and stored.
* Monitoring and Alerts: Real-time monitoring and alerting for issues or downtime will be available for administrators.

Security Requirements

* Data Encryption: Use SSL/TLS encryption for sensitive data, especially in workout and nutrition records.
* User Authentication: Implement strong authentication and optional MFA to secure user accounts.
* Authorization Controls: Role-based access ensures only administrators manage content, while users access their data.
* Input Validation: Validate all user inputs to prevent vulnerabilities like SQL injection and XSS.
* Session Management: Secure session management, using JWT tokens and timeout features, ensures safe user sessions.
* Regular Security Audits: Conduct periodic audits to identify and address vulnerabilities.
* Incident Response Plan: Prepare response protocols to mitigate security breaches effectively.

Maintainability Requirements

* Modular Architecture: A modular design allows for updates or replacements without affecting the entire application.
* Code Documentation: All modules will be documented to ensure clear guidance for future enhancements.
* Version Control: Use Git for tracking changes, managing versions, and collaboration.
* Testing Framework: Implement unit, integration, and end-to-end tests to ensure new updates don’t introduce bugs.
* Consistent Coding Standards: Adhere to coding standards like ESLint and Prettier for quality code.
* Dependency Management: Manage dependencies via npm or yarn to keep libraries secure and up-to-date.
* CI/CD Pipeline: Use CI/CD for automated testing and deployment, ensuring reliable updates.
* Error Logging and Monitoring: Error logging tools help capture runtime errors and performance issues quickly.

Portability Requirements

* Cross-Platform Compatibility: The app will run seamlessly on Windows, macOS, and Linux.
* Responsive Design: Using responsive design principles ensures optimal performance on mobile, tablet, and desktop.
* Web Standards Compliance: Follow HTML5, CSS3, and W3C standards for broad compatibility.
* Containerization: Use Docker for consistent deployment across development and production environments.
* Cloud Deployment: The app will be hosted on a scalable cloud platform, such as AWS or Google Cloud.
* API-First Approach: An API-first design allows easy integration with external services, like AI models.
* Minimal External Dependencies: Limit dependencies to avoid migration or upgrade issues.
* Documentation for Migration: Provide clear migration documentation for easier transfer to new environments.

Design Constraints

* Technology Stack: Use React.js for the frontend, PHP for backend APIs, MongoDB as the database, and AI APIs for fitness/nutrition recommendations.
* Browser Compatibility: Ensure compatibility with the latest Chrome, Firefox, Safari, and Edge versions.
* Performance Limitations: The application shall maintain a 2-second page load time for 500 concurrent users.
* Security Standards: Adhere to GDPR standards for data protection, as sensitive user data is involved.
* Responsive Design: Ensure responsive design for a consistent user experience across devices.
* Deployment Environment: Deploy on a cloud platform (e.g., AWS) for scalability and regular backups.
* Accessibility Requirements: Adhere to WCAG 2.1 to make the site accessible for users with disabilities.
* Third-Party Integrations: Integrate AI tools for fitness/nutrition insights and support email notifications.
* Localization: Support multi-language capabilities for international users.
* Version Control: Use Git for version control to maintain an organized and secure codebase.

Other Requirements

* User Interface Requirements: The interface should be clean, user-friendly, and maintain a consistent design across pages.
* Mobile Compatibility: Ensure mobile optimization for workout and nutrition logs, making key features accessible on smaller screens.
* Backup and Recovery Procedures: Schedule regular database backups and maintain documented recovery protocols.
* Documentation: Provide comprehensive user and technical documentation, including troubleshooting guidance.
* Compliance Requirements: Adhere to legal standards like GDPR for data privacy.
* Support and Maintenance: Establish a support plan, including a helpdesk or ticketing system, for technical assistance.
* Testing Requirements: Rigorous testing, including UAT, will ensure functionality and security before deployment.
* Monitoring and Analytics: Implement analytics for user behavior tracking, performance metrics, and error monitoring.
* Feedback Mechanism: Integrate a feedback system to capture user suggestions for continuous improvement.
* Integration with Analytics Tools: Connect to web analytics (e.g., Google Analytics) to track engagement and performance.

Analysis Models

* Use Case Diagrams: Show user interactions, such as setting workout/nutrition goals, generating recommendations, and administrators updating content.
* Activity Diagrams: Map workflows for setting and updating workout plans and AI-generated advice.
* Sequence Diagrams: Outline the interactions between components, particularly in AI recommendation flows and data retrieval.
* Entity-Relationship Diagram (ERD): Model relationships between users, workout plans, nutrition plans, and records in MongoDB.
* Data Flow Diagrams (DFD): Show data flow for user inputs, AI recommendations, and downloadable schedules.
* State Diagrams: Track state changes in workout/nutrition logs, AI chat status, and completed plans.
* Wireframes and Mockups: Create layout previews for key interfaces, focusing on a user-friendly experience.
* Functional Requirement Matrix: Map each functional requirement to its corresponding feature or component.

These analysis models will clarify the system’s functionality, user interactions, and data management, forming a foundation for development and deployment.Data Flow Diagrams (DFD)

Data Flow Diagram (DFD) for Fitness and Nutrition Planner Website

A Data Flow Diagram (DFD) visually represents how data moves through the Fitness and Nutrition Planner system. Below is a description of the main components and flows within a DFD for the Fitness and Nutrition Planner website.

Level 0 DFD (Context Diagram)

This diagram provides a high-level view of the system:

External Entities:

* User (Guest): Interacts with the system to provide workout preferences, retrieve nutrition plans, and chat with the AI assistant.
* Administrator: Manages user accounts, workout and nutrition resources, and oversees AI functionalities.

Processes:

* Fitness and Nutrition Planner System: Central system that processes user requests and manages data related to workouts, nutrition, and user interactions with AI.

Data Flows:

* User Input: From User to Fitness and Nutrition Planner System (e.g., workout preferences, nutritional goals).
* AI Recommendations: From Fitness and Nutrition Planner System to User (e.g., workout plans, meal suggestions).
* User Feedback: From User to Fitness and Nutrition Planner System (e.g., feedback on recommendations).
* Admin Requests: From Administrator to Fitness and Nutrition Planner System (e.g., updates to resources).
* Admin Data Management: From Fitness and Nutrition Planner System to Administrator (e.g., user statistics, system reports).

Level 1 DFD

This diagram breaks down the Fitness and Nutrition Planner System into more detailed processes:

Processes:

1. User Registration and Authentication

* Input: Registration data, login credentials.
* Output: Confirmation, error messages.

1. Workout and Nutrition Plan Retrieval

* Input: User preferences (e.g., fitness goals, dietary restrictions).
* Output: Customized workout and nutrition plans.

1. AI Interaction

* Input: User queries related to fitness and nutrition.
* Output: AI-generated responses and suggestions.

1. User Feedback Management

* Input: Feedback from users on workout/nutrition plans and AI recommendations.
* Output: Feedback summary for administrators, potential adjustments to recommendations.

1. Admin Management

* Input: Updates to user accounts, workout, and nutrition resources.
* Output: Confirmation of updates, user activity reports.

Data Stores:

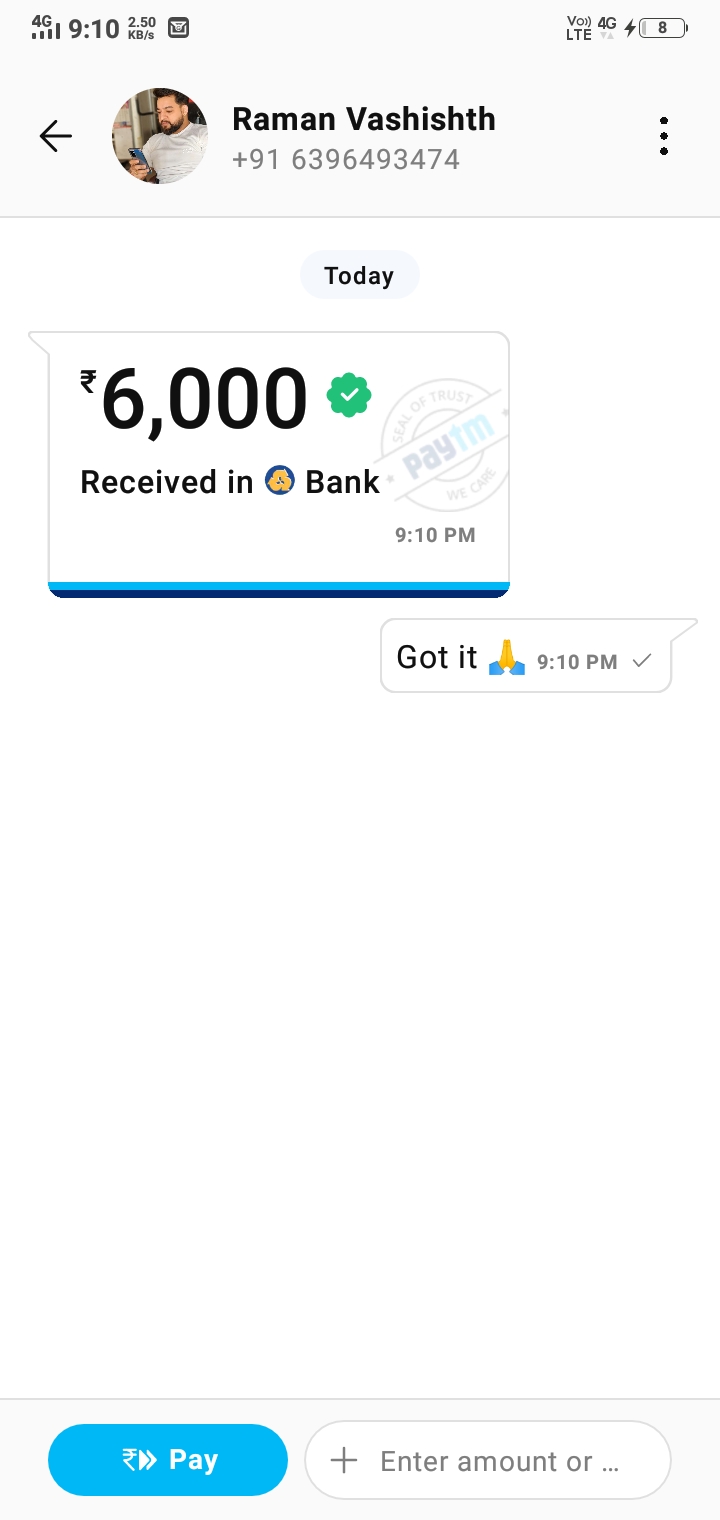
* User Database: Stores user information, authentication data, and preferences.
* Workout Database: Stores details of available workout plans and nutrition resources.
* AI Model Database: Contains information and configurations for the AI assistant’s functionalities.

Data Flows in Level 1 DFD

* User Registration Data: Flows from User to User Registration and Authentication.
* User Credentials: Flows from User to User Registration and Authentication for login.
* User Preferences: Flows from User to Workout and Nutrition Plan Retrieval.
* Customized Plans: Flows from Workout and Nutrition Plan Retrieval to User.
* User Queries: Flows from User to AI Interaction.
* AI Responses: Flows from AI Interaction to User.
* User Feedback: Flows from User to User Feedback Management.
* Feedback Summary: Flows from User Feedback Management to Admin Management.
* Admin Updates: Flows from Administrator to Admin Management.
* Updated Data: Flows from Admin Management to User Database and Workout Database.

This DFD captures the flow of information and processes within the Fitness and Nutrition Planner system, illustrating how users and administrators interact with the system and how data is managed throughout different functionalities. It focuses on the integration of user inputs, AI recommendations, and administrative controls, ensuring a comprehensive understanding of the system’s operations.

**TRANSACTION FROM THE CLIENT :**



**Client ID Proof->**

****

**Client location : Hathras, Uttar Pradesh**



**Github link->**

**<https://github.com/07-utkarsh/Rg_fitness_club/tree/main/gym-website-main>**

**Deployed link->**