

Software Engineering Software Requirements Specification (SRS) Document

MUSCLE MENTOR

2/12/2024

[Version 1]

By: Sara Abukhalaf, Amna Sohail,
Fady Eskandr, Bunroung Heak

WE HAVE ABIDED BY THE UNCG ACADEMIC POLICY ON THIS ASSIGNMENT.

1.	Introduction	3
1.1.	Purpose	3
1.2.	Document Conventions	3
1.3.	Definitions, Acronyms, and Abbreviations	4
1.4.	Intended Audience	5
1.5.	Project Scope	5
1.6.	Technology Challenges	5
1.7.	References	6
2.	General Description	7
2.1.	Product Features	7
2.2.	User Class and Characteristics	7
2.3.	Operating Environment	7
2.4.	Constraints	7
2.5.	Assumptions and Dependencies	7
3.	Functional Requirements	8
3.1.	Primary	8
3.2.	Secondary	8
3.3.	Use-Case Model	9
3.3.1.	Use-Case Model Diagram	9
3.3.2.	Use-Case Model Descriptions	9
3.3.2.1.	Actor: Manager (Alice)	9
3.3.2.2.	Actor: Actor Name (Responsible Team Member)	9
3.3.2.3.	Actor: Actor Name (Responsible Team Member)	10
3.3.3.	Use-Case Model Scenarios	10
3.3.3.1.	Actor: Manager (Alice)	10
3.3.3.2.	Actor: Actor Name (Responsible Team Member)	11
3.3.3.3.	Actor: Actor Name (Responsible Team Member)	11
4.	Technical Requirements	12

8.1. Interface Requirements	12
8.1.1.	User Interfaces
12	
8.1.2.	Hardware Interfaces
13	
8.1.3.	Communications Interfaces
13	
8.1.4.	Software Interfaces
13	
9. Non-Functional Requirements	14
9.1. Performance Requirements	14
9.2. Safety Requirements	14
9.3. Security Requirements	14
9.4. Software Quality Attributes	14
9.4.1.	Availability
14	
9.4.2.	Correctness
14	
9.4.3.	Maintainability
14	
9.4.4.	Reusability
14	
9.4.5.	Portability
14	
9.5. Process Requirements	14
9.5.1.	Development Process Used
14	
9.5.2.	Time Constraints
14	
9.5.3.	Cost and Delivery Date
15	
9.6. Other Requirements	15
10. Design Documents	11
10.1. Software Architecture	11
10.2. High-Level Database Schema	11
10.3. Software Design	11
10.3.1. State Machine Diagram: Actor Name (Responsible Team Member)	11

10.3.2. State Machine Diagram: Actor Name (Responsible Team Member)	11
10.3.3. State Machine Diagram: Actor Name (Responsible Team Member)	11
10.4. UML Class Diagram	11
11. Scenario	11
11.1. Brief Written Scenario with Screenshots	11

1. Introduction

1.1. Purpose

- The purpose of this Software Requirements Document (SRD) is to outline the features and functions of our fitness tracking and workout planning software. The primary goal is to provide users with a comprehensive tool to track their weight, calculate Body Mass Index (BMI), and allow each user to choose a personalized fitness journey. The software aims to cater to users with diverse fitness goals, offering different workout plans based on their gym goals, whether it's muscle gain or weight loss.

1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to provide a comprehensive and detailed outline of the requirements, features, and functions of our fitness tracking and workout planning software. This document serves as a foundational guide for the development team, project managers, and others involved in the project.

In it, we will have:

- **Weight Tracking:** Allows each user to log on and monitor their weight progress over time.
- **BMI Calculator:** Provide a BMI calculation feature to assess the user's body mass index.
- **Personalized Workout Plans:** Generate daily workout plans assigned to each user's fitness goals, whether it's muscle gain or weight loss.
- **Progress Tracking:** Implement a system to track and provides a display of user progress over time.
- **Motivational Incentives:** Encourage regular software usage by rewarding users with coupon codes for healthy or high-protein meals or encouraging quites that cycle throughout each login.

1.3. Definitions, Acronyms, and Abbreviations

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Restaurant Manager.
MySQL	Open-source relational database management system.
.HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
Spring Web	Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
NetBeans	An integrated development environment (IDE) for Java. This is where our system will be created.
API	Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage.

BMI	Body Mass Index
-----	-----------------

1.4. Intended Audience

- This Software Requirements Document (SRD) is designed to serve to a diverse audience, each with specific roles and responsibilities in the project. Each audience member should be able to find conclusive evidence on resources and integration requirements. One stakeholder will be **Developers**, which should be able to find detailed information on the functions to be implemented, technical specifications, and the overall system design, which includes, details on programming languages, frameworks, and tools to be utilized. Another are **Project Managers**. Project managers should be able to find insight to overall goals of the software, alongside any constraints. This as well includes the scope of the project, any potential challenges that can impact project timelines, and any benefits to the business. The **User** is another stakeholder since this document should allow the user to find features on the application as well as features that directly impact them. Examples of this can include BMI calculation, Progress tracking, workout plans, and persuasive incentives. Since there can be a large audience in the fitness industry, the Software Requirements Document (SRD) can be useful for a **Quality Assurance Team** as well. QA teams will use this document to understand the expected behavior of the software, including functional requirements.

1.5. Project Scope

- The goals of the software align with the broader business objectives of promoting user health and fitness. The scope of the project is defined by its key features like Weight tracking, a BMI calculator, Personal workout plans, and progress tracking. These are all motivational incentives to help provide a competitive edge in the fitness application industry. The software's features encourage users to regularly engage with the platform by providing these key features. By Offering coupon codes as well for healthy meals aligns with the business's emphasis on promoting a healthy lifestyle. By providing the business with a competitive edge by addressing users' fitness goals, the software goals and the benefits to the business include improved user engagement, customer satisfaction, and promotion of a healthy lifestyle which can all increase brand loyalty.

1.6. Technology Challenges

- Throughout the development of Muscle Mania, there may be a few technological constraints. For example, one of the most important constraint/challenge will be the implementation of the weight tracker API. Integrating a reliable and accurate Weight Tracking API for seamless weight data input is one of the most important goals but can be one of the main challenges. Another constraint may be creating a diverse workout plan that is personalized to each user however a solution may be to collaborate with fitness experts to design effective workout algorithms. A huge challenge that the project will face will be its security, and safeguarding user data, especially considering the sensitive nature of health and fitness information. Finally, implementing a working coupon code system that encourages users to use the software regularly. Finally, a constraint that this software can face in the future can be scalability. Ensuring the software can handle a growing user base without sacrificing performance can be a large challenge this software faces.

1.7. References

<https://rapidapi.com/CTRO/api/calorie-calculator/> -- Calorie calculator API

<https://api-ninjas.com/api/exercises> -- Exercises API

2. General Description

2.1. Product Features

The product features of Muscle Mentor include user registration for fitness enthusiasts and trainers, with administrative tools for account management. Users can log their weight, track their BMI, and begin on personalized fitness journeys tweaked to their goals. The app offers workout plans based on individual preferences, whether it's muscle gain or weight loss, with varying intensity levels. Also, regular users receive rewards such as coupon codes for healthy meals, and promotions to their fitness goals.

2.2. User Class and Characteristics

Our website application expects the user to have prior knowledge of a computer and calculations because it is needed in order to give the client/customer the appropriate fitness and workout plan. Our website application will ask the client for their measurements in regarding to health so we can take this information and provide them with what they are asking for like meal plans, workout plans, advice etc to help them gain the best experience possible.

2.3. Operating Environment

The application is designed to operate on the web across many different devices like phones, computers, watches, anything that can track you.

2.4. Constraints

For design Constraints----Data privacy and security, violations of HIPPA or GDPR for example will be a challenge because of the sensitive information that will be given to us, so we can provide a link to where the client will sign and consent to the information they are giving out and who will be accessing that information in the backend (us). Also gathering real time data will be a challenge since this application will be available on all tracking platforms.

2.5. Assumptions and Dependencies

The software will be dependent on html and c/c++ and visual studio code, also java (because some of us dont know how to code in a similar language proffeciantly) Also mySQL and spring web (we are still deciding on everything we want to use because we are testing the waters)

3. Functional Requirements

3.1. Primary

- FR0: The app will allow users to create a personal profile to track their fitness journey. Users can input their personal information such as age, weight, height, fitness goals, and any health conditions.

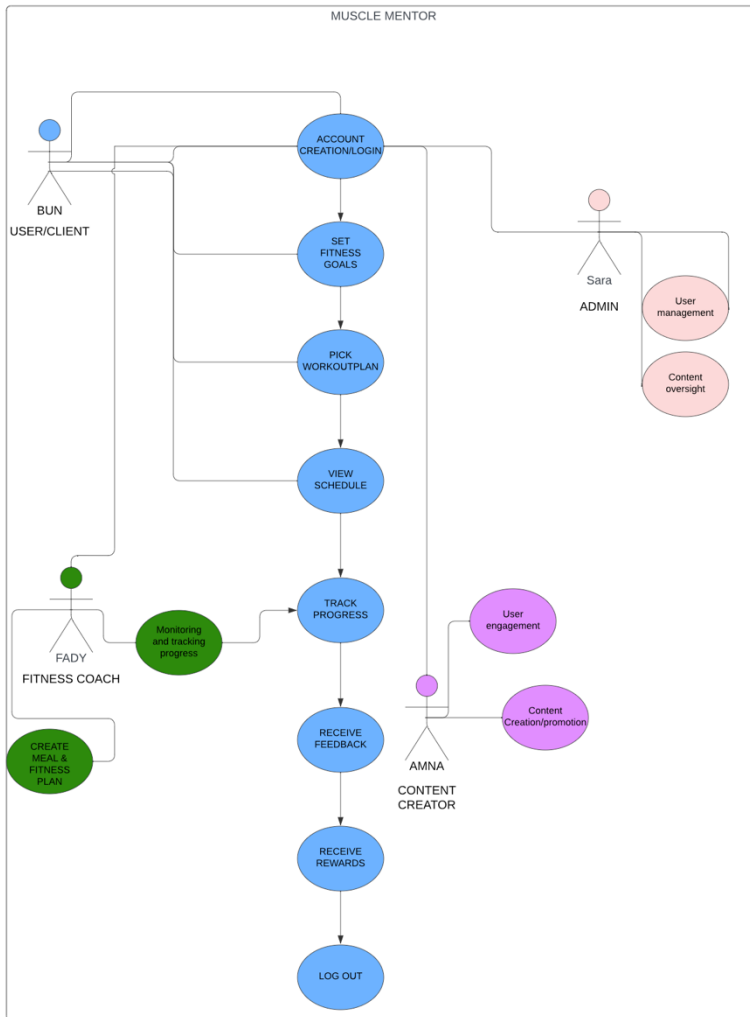
FR1: The app will calculate the (BMI) of users based on their height and weight input. Users can view their BMI score and understand their current health status based on the BMI categories (e.g., underweight, normal weight, overweight, obese).
- FR2: The app will offer users personalized workout plans matched to their fitness goals and preferences. Users can specify whether they want to gain muscle mass, lose weight, or improve overall fitness. The app will generate daily workout routines consisting of exercises, sets, reps, and rest intervals from our fitness coach (Fady).
- FR4: The app will track users' progress in completing their workout plans. Users can log their workouts, including exercises performed, weights lifted, and repetitions completed. The app will display progress metrics such as workout duration, calories burned, and muscle groups targeted.
- FR4: The app will provide users with nutritional guidance and meal planning features. Users can access a database of healthy recipes and meal suggestions based on their dietary preferences, calorie intake goals, and macronutrient requirements to match their fitness goal/plan.

3.2. Secondary

- **User-friendly Authentication Interface:** an intuitive interface for users to log in with their credentials easily.
- **Personalized Workout Access:** Ensure users can only view and modify their own workout plans and progress.
- **Role Definitions:** Defining fundamental roles for trainers, administrators, and users with clear permissions.
- **Authorization scheme so that customers can only alter and see their information, tracking, meal plans and workout plans etc.**

3.3. Use-Case Model

3.3.1. Use-Case Model Diagram



3.3.2. Use-Case Model Descriptions

3.3.2.1. Actor: Adminastrator (Sara)

- **User Management:** [can manage user accounts, including creating new accounts, updating existing ones, and deactivating or deleting accounts as needed]
- **Content Oversight:** [can oversee and manage the content within the system, including approving or rejecting user-generated content, monitoring activity logs, and ensuring compliance with content guidelines and policies]

3.3.2.2. Actor: Fitness Coach (Fady)

- **Monitor and Track Progress:** [can monitor and track the progress of clients, including assessing their performance, tracking completed workouts, and recording any changes in weight or measurements over time.]
- **Create Meal and Fitness Plans:** [can create personalized meal and fitness plans for clients based on their goals, preferences, and dietary requirements. This includes designing workout routines and recommending nutrition plans to help clients achieve their fitness objectives.]

3.3.2.3. Actor: Content Creator (Amna)

- **User Engagement:** [providing motivation and inspiration to users through visually captivating designs, motivational quotes, success stories, and interactive features like promotions aimed at encouraging users to stay committed to their fitness journey and achieve their goals.]
- **Content Creation and Promotion:** [can create and promote content within the application, including designing graphics, writing articles or blog posts, and utilizing social media channels to increase visibility and attract users to the platform.]

4.4.2.4. Actor: User/Client (Bun)

- **User Journey:** [can manage their entire journey within the application, starting from logging in to accessing personalized plans/meals and receiving rewards, promotions and ending to logging out and repeating as much as desired.]

3.3.3. Use-Case Model Scenarios

3.3.3.1. Actor: Administrator (Sara)

- **Use-Case Name: User Management**
 - **Initial Assumption:** Has access to the webapp and holds administrative privileges.
 - **Normal:** creates a new user account by filling in the required information, updating an existing user's details, or deactivating a user account as needed.
 - **What Can Go Wrong:** Admin tries to delete a user account that still has pending tasks assigned to it, The system should prevent the deletion and provide an error message with the reason for it.
 - **Other Activities:** monitoring user account logs to ensure policy compliance.
 - **System State on Completion:** The user accounts are managed according to the actions taken by Sara, and any changes are reflected in the system.
- **Use-Case Name: Content Oversight**
 - **Initial Assumption:** The administrator has access to content management tools and can oversee all content within the system.
 - **Normal:** Admin reviews user-generated content submissions, approves those that meet content guidelines, and denies other content for further review.
 - **What Can Go Wrong:** Admin accidentally approves a piece of content that violates content guidelines. The system should allow her to revoke approval and correct it.
 - **Other Activities:** Admin can monitor user activity logs to detect any suspicious behavior or content violations and takes appropriate action.
 - **System State on Completion:** The content within the system is effectively overseen and managed, ensuring compliance with content policies and guidelines.

3.3.3.2. Actor: Actor Fitness Coach (Fady)

- **Use-Case Name: Monitor and Track Progress**
 - **Initial Assumption:** Coach has access to client profiles and progress tracking within the application.
 - **Normal:** Assesses a client's performance, tracks completed workouts, and records any changes in weight or measurements over time.
 - **What Can Go Wrong:** Coach encounters a technical issue preventing him from accessing a client's progress data.
 - **Other Activities:** Coach communicates with clients to provide feedback on their progress and adjust their fitness plans accordingly.
 - **System State on Completion:** The progress of clients is accurately monitored and tracked, and Coach has the necessary information to support their fitness journey effectively.
- **Use-Case Name: Create Meal and Fitness Plans**
 - **Initial Assumption:** Coach has access to tools and resources for designing personalized meal and fitness plans within the application.
 - **Normal:** Coach creates customized meal plans and workout routines for clients based on their goals, preferences, and dietary requirements.
 - **What Can Go Wrong:** Coach accidentally assigns an unsuitable workout routine to a client with specific health considerations. The system should provide warnings or prompts to ensure Fady selects appropriate plans for each client.
 - **Other Activities:** Coach regularly reviews and updates client plans based on their progress and feedback.
 - **System State on Completion:** Clients have personalized meal and fitness plans tweaked to their needs, helping them achieve their fitness objectives effectively.

3.3.3.3. Actor: Actor Name Content Creator (Amna)

- **Use-Case Name: User Engagement**
 - **Initial Assumption:** Creator has access to content creation tools and engagement features within the application.
 - **Normal:** Creator creates visually appealing content, motivational quotes, and interactive features aimed at inspiring and motivating users to stay committed to their fitness journey!
 - **What Can Go Wrong:** Creator encounters difficulties in uploading content due to technical issues with the platform.
 - **Other Activities:** Creator monitors user engagement metrics and adjusts content strategies based on user feedback and engagement levels.
 - **System State on Completion:** Users are actively engaged with the platform through great content and interactive features, contributing to their motivation and success in achieving their fitness goals.
- **Use-Case Name: Content Creation and Promotion**
 - **Initial Assumption:** Creator has access to content creation and promotion tools, including social media integration, within the application.

- **Normal:** Creator creates and promotes content such as graphics, articles, and blog posts within the application, utilizing various outlets to increase visibility and attract users to the platform.
- **What Can Go Wrong:** Promotional campaigns do not live up to the expected results due to ineffective messaging or targeting.
- **Other Activities:** Creator collaborates with other team members to align content creation and promotion efforts with overall marketing goals and strategies.
- **System State on Completion:** The platform's content is effectively created, promoted, and distributed across various outlets, contributing to increased user engagement and platform growth.

3.3.3.4. Actor: Actor Name User/Client (Bun)

- Use-Case Name: User Engagement

- **Initial Assumption:** Bun has registered an account and can access the full range of features within the application.
- **Normal:** Bun logs in to the application, accesses personalized meal and fitness plans, tracks progress, and engages with motivational content and interactive features.
- **What Can Go Wrong:** Bun encounters difficulties accessing their personalized plans or experiences errors while tracking progress due to connectivity issues.
- **Other Activities:** Bun participates in challenges, earns rewards, and interacts with other users within the community to stay motivated and committed to their fitness journey and reaches their fitness goals through personalized dietary meals and workouts.
- **System State on Completion:** Bun's user journey within the application is fulfilling, leading to improved engagement and progress towards fitness goals.

4. Technical Requirements

4.1. Interface Requirements

4.1.1. User Interfaces

The app will feature a user-friendly interface with a clean and intuitive design. Screens will be organized to provide easy navigation, with dedicated sections for workouts, nutrition, progress tracking, etc. We will put emphasis on visually appealing graphics to enhance user engagement. There will be appropriate placement of essential buttons, such as "Change Weight," and "View Progress," to ensure user-friendliness. Special attention will be paid to a vibrant color scheme to reflect the app's fitness theme. There will also be clear prompts for any user actions, such as successful log entries or goal achievements.

4.1.2. Hardware Interfaces

The web application will be designed to operate on diverse hardware devices with internet connectivity and the ability to interact with the page. Muscle Mentor will be accessible for a wide range of devices. This includes:

Smartphones (iOS and Android platforms).

Tablets (iOS and Android platforms).

Desktop computers (Windows and macOS).

Laptops (Windows and macOS).

The app requires a stable internet connection for real-time data updates.

4.1.3. Communications Interfaces

The communication protocol HTTP is used to connect to the World Time API.

The app examines the API to retrieve the current date and time information, enhancing user experiences by providing accurate timestamps. Muscle Mentor relies on HTTP and HTTPS protocols to connect to the internet.

4.1.4. Software Interfaces

Muscle Mentor may unite with external fitness APIs to get additional data for exercises, nutrition, and other relevant information. This integration can enhance the app's content and provide users with a better fitness experience.

5. Non-Functional Requirements

5.1. Performance Requirements

- NFR0(R): The local copy of the user data and work out plan will consume less than 100MB of memory.
- NFR1(R): The usage of memory for BMI calculation and other calculations methods will consume less than 100MB.
- NFR2(R): The entire program will consume less than 200MB of memory
- NFR3(R): The interface should be simple enough to where it will take less than 10 minutes to create a new profile and custom workout.
- NFR4(R): Reacurring users should be able to start a new workout in less than 5 minutes.
- NFR5(R): The interface should allow the user to create a new workout routine in less than 10 minutes.

5.2. Safety Requirements

- NFR0(R): The progress of the user will be stored locally to help safeguard user data
- NFR1(R): The program will have proper handling error that will not expose private information when there is a failure
- NFR2(R): The program will regularly back up to prevent loss of data incase of a crash.

5.3. Security Requirements

- NFR0(R): The system will need user to enter in a username and password.
- NFR1(R): The system will rely on a two factor authenticator to veriy the user.
- NFR2(R): The system will run on an custom API.
- NFR3(R): The system will timeout the user that request too many failed login
- NFR4(R): Add encrypted data to the app.

5.4. Software Quality Attributes

5.4.1. Availability

The program will be on a webpage but hopefully we will be able to create an app ported for phones.

5.4.2. Correctness

5.4.3. Maintainability

The code should be detailed in comments to ensure ease of debugging and maintaince.

5.4.4. Reusability

The interface, ease of use, daily task ,app design and the vast amount of resources should keep user on track to their goals with daily use.

5.4.5. Portability

The program will be first ported to HTML page and a website will be created housing all of the app features.

5.5. Process Requirements

5.5.1. Development Process Used

5.5.2. Time Constraints

The app will start it's production late Febuary and it will be completed by early May

5.5.3. Cost and Delivery Date

The expected delivery date would be May.

5.6. Other Requirements