Fitfluencer App

Elevate your fitness journey with Fitfluencer: Where inspiration meets perspiration.

Table of Contents

[1. Executive Summary The FitFluencer app is designed to integrate fitness enthusiasts with influencers through a platform that combines social networking and personalized fitness guidance. Developed by a team of software engineers, UI/UX designers, and project managers, the app targets users seeking customized workout plans and influencers wanting to engage their audience. This document provides a detailed project plan including system architecture, design specifications, and development timeline. To date, we have completed market research, initial design, and core feature development. The project is in the specification phase, with the next steps involving detailed usability testing, finalizing features, and preparing for deployment. Our focus is on delivering a technically robust and user-friendly fitness application. 5](#_Toc168577100)

[2. Introduction 6](#_Toc168577101)

[2.1. Problem Statement / Project Vision 6](#_Toc168577102)

[2.2. System Capabilities 7](#_Toc168577103)

[ Functional Requirements: 7](#_Toc168577104)

[a) News Feed/Activity Stream: Users can view updates and posts from fitness influencers they follow, providing a curated feed of relevant content. 7](#_Toc168577105)

[b) Direct Messaging Feature: Users can communicate directly with subscribed fitness influencers on a one-on-one basis, fostering personalized interactions and support. 7](#_Toc168577106)

[c) Personal Profiles for Fitness Influencers: Each fitness influencer has a dedicated profile showcasing their content, expertise, and contributions to the app community, allowing users to explore their offerings easily. 7](#_Toc168577107)

[d) Content Moderation: Strict moderation policies and tools are in place to ensure that the platform remains free of unwanted or inappropriate content, maintaining a safe and positive user experience. 7](#_Toc168577108)

[e) Desktop Website: The app offers a desktop website counterpart, extending its reachability to users who prefer to access the platform via traditional web browsers. 7](#_Toc168577109)

[f) Macro Tracking and Gym Progress Recording: Additional features include tools for tracking macronutrient intake and recording progress in the gym, providing users with comprehensive health and fitness management capabilities. 7](#_Toc168577110)

[ Use Case Descriptions: 7](#_Toc168577111)

[a) Create/Publish Content (UC-1): This use case involves actions performed by a content creator to generate and publish fitness-related content for subscribers. The process includes accessing the content creation interface, incorporating multimedia elements and finally publishing the content. For more detail, refer to Section 4 and the Create/Publish Content Use Case. 7](#_Toc168577112)

[b) Subscribe to Fitness Plans (UC-2): Describes how a user subscribes to fitness plans within the Fitfluencer application, providing access to curated fitness content tailored to their goals. For more detail, refer to Section 4 and the Subscribe to Fitness lans Use Case. 7](#_Toc168577113)

[c) Manage User Accounts/Moderate Content (UC-3): Involves an admin managing user accounts and moderating content to maintain a safe and compliant environment within the Fitfluencer app. For more detail, refer to Section 4 and the Manage User Accounts/Moderae Content Use Case. 7](#_Toc168577114)

[2.3. Non-functional Requirements and Design Constraints 8](#_Toc168577115)

[Constraints: 8](#_Toc168577116)

[a) Technical Compatibility: Ensuring compatibility across devices and operating systems may impact project timelines; we'll prioritize popular platforms and conduct thorough testing (Section 1). 8](#_Toc168577117)

[b) Data Privacy and Security: Strict regulations demand robust security measures, potentially increasing costs; encryption, compliance efforts, and transparent policies will be implemented (Section 1). 8](#_Toc168577118)

[c) Scalability: Seamless scaling is crucial; scalable infrastructure and modular design will address potential issues (Section 1). 8](#_Toc168577119)

[d) Resource Constraints: Limited budget and resources may affect development scope and pace; prioritization and agile methodologies will maximize efficiency (Section 1). 8](#_Toc168577120)

[Non-functional Requirements: 8](#_Toc168577121)

[a) Performance: Fast loading times and responsive interactions are essential (Section 4). 8](#_Toc168577122)

[b) Security: Encryption and secure authentication are imperative for user data protection (Section 4). 8](#_Toc168577123)

[c) Usability: An intuitive interface and accessibility features are required (Section 4). 8](#_Toc168577124)

[d) Reliability: High availability and backup mechanisms are essential (Section 4). 8](#_Toc168577125)

[e) Scalability: Capacity to handle increased loads and feature expansions is necessary (Section 4). 8](#_Toc168577126)

[f) Compliance: Adherence to data protection regulations and content moderation is vital (Section 4). 8](#_Toc168577127)

[g) Compatibility: Support for diverse devices, operating systems, and web browsers is required (Section 4). 8](#_Toc168577128)

[Feasibility Analysis: 8](#_Toc168577129)

[a) Technical Feasibility (Moderate): Integrating complex features may pose challenges despite available technologies (Section 4). 8](#_Toc168577130)

[b) Resource Feasibility (Low): Adequate resources are allocated, but careful management and potential outsourcing may be necessary (Section 4). 8](#_Toc168577131)

[c) Schedule Feasibility (Low): Project timeline is reasonable, but agile methodologies are crucial to mitigate risks (Section 4). 8](#_Toc168577132)

[d) Organizational Feasibility (High): Stakeholder support aligns with organizational goals (Section 4). 8](#_Toc168577133)

[e) Legal and Contractual Feasibility (Moderate): Compliance with laws and contractual agreements is vital; careful review and negotiation are necessary (Section 4). 8](#_Toc168577134)

[2.4. System Evolution 9](#_Toc168577135)

[2.5. Document Outline 10](#_Toc168577136)

[3. Structural Model 11](#_Toc168577137)

[3.1. Model Introduction 11](#_Toc168577138)

[3.2. Class Diagrams 11](#_Toc168577139)

[3.3. Metadata 11](#_Toc168577140)

[4. Architecture Design 12](#_Toc168577141)

[4.1. Architecture Overview 12](#_Toc168577142)

[4.2. Infrastructure Model 12](#_Toc168577143)

[**4.2.1.** **Deployment Diagram 1 – Architecture Overview** 12](#_Toc168577144)

[**4.2.2.** **Deployment Diagram 2 – Nodes and Artifacts** 12](#_Toc168577145)

[4.3. Hardware and Software Requirements 13](#_Toc168577146)

[**4.3.1.** **Hardware Components** 13](#_Toc168577147)

[ For the FitFluencer app's architecture, we'll need: 13](#_Toc168577148)

[a) Web Servers: Hosting the web version. 13](#_Toc168577149)

[b) Application Servers: For backend logic and APIs. 13](#_Toc168577150)

[c) Database Servers: Storing user data. 13](#_Toc168577151)

[d) Load Balancers: Distributing web and app traffic. 13](#_Toc168577152)

[ Recommendation: Invest in new hardware for optimal performance. Users will need web browsers for the web version and iOS/Android devices for the respective applications. 13](#_Toc168577153)

[**4.3.2.** **Required Software Components** 13](#_Toc168577154)

[4.4. Security Plan 13](#_Toc168577155)

[**4.4.1.** **Security Overview** 13](#_Toc168577156)

[**4.4.2.** **Security Plan** 14](#_Toc168577157)

[**Threat Category** 14](#_Toc168577158)

[**Potential Threats** 14](#_Toc168577159)

[**Controls** 14](#_Toc168577160)

[Physical Security 14](#_Toc168577161)

[Theft or damage to server hardware 14](#_Toc168577162)

[Implement physical access controls and surveillance 14](#_Toc168577163)

[Network Security 14](#_Toc168577164)

[Unauthorized access to user data during transmission 14](#_Toc168577165)

[Encrypt data transmission using SSL/TLS protocols 14](#_Toc168577166)

[Application Security 14](#_Toc168577167)

[SQL injection attacks 14](#_Toc168577168)

[Implement input validation and parameterized queries 14](#_Toc168577169)

[File Security 14](#_Toc168577170)

[Unauthorized access to stored user data 14](#_Toc168577171)

[Implement file encryption and access control measures 14](#_Toc168577172)

[User Security 14](#_Toc168577173)

[Weak user passwords 14](#_Toc168577174)

[Enforce password complexity requirements and 2FA 14](#_Toc168577175)

[5. User-Interface 15](#_Toc168577176)

[5.1. User-Interface Requirements and Constraints 15](#_Toc168577177)

[5.2. Window/Screen Navigation Diagram 15](#_Toc168577178)

[5.3. UI Wireframes 16](#_Toc168577179)

[5.4. Reports: "Formal Output" Design 18](#_Toc168577180)

[6. Appendices 19](#_Toc168577181)

[6.1. Glossary 19](#_Toc168577182)

[6.2. References / Bibliography 20](#_Toc168577183)

[6.3. Supporting documentation 20](#_Toc168577184)

# **Executive Summary** The FitFluencer app is designed to integrate fitness enthusiasts with influencers through a platform that combines social networking and personalized fitness guidance. Developed by a team of software engineers, UI/UX designers, and project managers, the app targets users seeking customized workout plans and influencers wanting to engage their audience. This document provides a detailed project plan including system architecture, design specifications, and development timeline. To date, we have completed market research, initial design, and core feature development. The project is in the specification phase, with the next steps involving detailed usability testing, finalizing features, and preparing for deployment. Our focus is on delivering a technically robust and user-friendly fitness application.

# **Introduction**

The FitFluencer app connects fitness enthusiasts with influencers, offering a blend of social networking and personalized fitness guidance. Users can create profiles, access customized workout plans, track progress, and engage with a community. Influencers can share content, interact with followers, and offer training programs. Key requirements include:

1. User Profiles: For both enthusiasts and influencers.
2. Workout Plans: Customizable routines with video demos.
3. Social Networking: Follow, like, comment, and share.
4. Progress Tracking: Log workouts and visualize progress.
5. Notifications: Updates on new content, workouts, and milestones.
6. Security and Privacy: Secure authentication and data protection.

## **Problem Statement / Project Vision**

The FitFluencer app arises from the recognition of a widespread challenge among individuals striving for fitness goals: the lack of personalized guidance, support systems, and accountability measures. This problem statement catalyzes the project's vision to revolutionize personal fitness journeys. FitFluencer aims to fill this gap by providing a comprehensive platform offering tailored workout plans, nutrition guidance, progress tracking, and community support, ultimately fostering healthier lifestyles among its user base.

The project's vision is to create an innovative mobile app that empowers users to achieve health and wellness goals through personalized workouts, tailored meal plans, and a supportive community, fostering a culture of motivation, accountability, and sustainable lifestyle changes. The scope encompasses features like personalized routines, meal plans, progress tracking, and social engagement tools across iOS and Android platforms. Stakeholders, ranging from fitness trainers to investors, share a common interest in a high-quality product that effectively communicates its value proposition, resonates with target audiences, and drives user acquisition and retention. Additionally, there's an emphasis on data privacy and security, compliance with relevant regulations, and aligning with evidence-based practices to ensure trust and credibility among users and regulatory authorities alike.

## **System Capabilities**

## Functional Requirements:

## News Feed/Activity Stream: Users can view updates and posts from fitness influencers they follow, providing a curated feed of relevant content.

## Direct Messaging Feature: Users can communicate directly with subscribed fitness influencers on a one-on-one basis, fostering personalized interactions and support.

## Personal Profiles for Fitness Influencers: Each fitness influencer has a dedicated profile showcasing their content, expertise, and contributions to the app community, allowing users to explore their offerings easily.

## Content Moderation: Strict moderation policies and tools are in place to ensure that the platform remains free of unwanted or inappropriate content, maintaining a safe and positive user experience.

## Desktop Website: The app offers a desktop website counterpart, extending its reachability to users who prefer to access the platform via traditional web browsers.

## Macro Tracking and Gym Progress Recording: Additional features include tools for tracking macronutrient intake and recording progress in the gym, providing users with comprehensive health and fitness management capabilities.

## Use Case Descriptions:

## Create/Publish Content (UC-1): This use case involves actions performed by a content creator to generate and publish fitness-related content for subscribers. The process includes accessing the content creation interface, incorporating multimedia elements, and finally publishing the content. For more detail, refer to Section 4 and the Create/Publish Content Use Case.

## Subscribe to Fitness Plans (UC-2): Describes how a user subscribes to fitness plans within the Fitfluencer application, providing access to curated fitness content tailored to their goals. For more detail, refer to Section 4 and the Subscribe to Fitness Plans Use Case.

## Manage User Accounts/Moderate Content (UC-3): Involves an admin managing user accounts and moderating content to maintain a safe and compliant environment within the Fitfluencer app. For more detail, refer to Section 4 and the Manage User Accounts/Moderate Content Use Case.

## **Non-functional Requirements and Design Constraints**

## Constraints:

## Technical Compatibility: Ensuring compatibility across devices and operating systems may impact project timelines; we'll prioritize popular platforms and conduct thorough testing (Section 1).

## Data Privacy and Security: Strict regulations demand robust security measures, potentially increasing costs; encryption, compliance efforts, and transparent policies will be implemented (Section 1).

## Scalability: Seamless scaling is crucial; scalable infrastructure and modular design will address potential issues (Section 1).

## Resource Constraints: Limited budget and resources may affect development scope and pace; prioritization and agile methodologies will maximize efficiency (Section 1).

## Non-functional Requirements:

## Performance: Fast loading times and responsive interactions are essential (Section 4).

## Security: Encryption and secure authentication are imperative for user data protection (Section 4).

## Usability: An intuitive interface and accessibility features are required (Section 4).

## Reliability: High availability and backup mechanisms are essential (Section 4).

## Scalability: Capacity to handle increased loads and feature expansions is necessary (Section 4).

## Compliance: Adherence to data protection regulations and content moderation is vital (Section 4).

## Compatibility: Support for diverse devices, operating systems, and web browsers is required (Section 4).

## Feasibility Analysis:

## Technical Feasibility (Moderate): Integrating complex features may pose challenges despite available technologies (Section 4).

## Resource Feasibility (Low): Adequate resources are allocated, but careful management and potential outsourcing may be necessary (Section 4).

## Schedule Feasibility (Low): Project timeline is reasonable, but agile methodologies are crucial to mitigate risks (Section 4).

## Organizational Feasibility (High): Stakeholder support aligns with organizational goals (Section 4).

## Legal and Contractual Feasibility (Moderate): Compliance with laws and contractual agreements is vital; careful review and negotiation are necessary (Section 4).

## 

## **System Evolution**

* 1. Social Integration: Enable users to share their fitness achievements and progress on social media platforms, enhancing engagement and motivation.
  2. Advanced Analytics: Introduce detailed analytics tools to provide deeper insights into fitness progress and habits, including predictive analytics for personalized recommendations.
  3. Community Challenges: Introduce group challenges and competitions to foster a sense of community and encourage active engagement with the app.
  4. Nutrition Tracking: Add features for tracking dietary intake and providing personalized nutrition plans to complement fitness goals.

In addition to these future features, planned upgrades include:

1. Hardware: Upgrade servers and storage solutions to ensure scalability and accommodate the growing user base, ensuring smooth and efficient app performance.
2. Software: Implement regular software updates to improve performance, enhance security, and introduce new features. Integration of machine learning algorithms will provide personalized user experiences based on individual fitness data and preferences.
   * 1. **Version 2 Changes**

* Use Case: Social Integration (UC-4):

Description: Users share fitness achievements on social media directly from the app.

Importance: High

* Use Case: Advanced Analytics (UC-5):

Description: Users gain deeper insights into their fitness progress with sophisticated analysis tools.

Importance: High

* Use Case: Community Challenges (UC-6):

Description: Engage users with group challenges and competitions to foster motivation and interaction.

Importance: Medium

* + 1. **Version 3 and beyond Changes**
* Use Case: Personalized Recommendations (UC-7):

Description: Provide tailored workout routines, nutrition plans, and content suggestions based on user preferences and past activities.

Importance: High

* Use Case: Virtual Fitness Classes (UC-8):

Description: Offer live or pre-recorded virtual fitness classes led by certified trainers.

Importance: Medium

* Use Case: Health Tracking Integration (UC-9):

Description: Sync data from wearable fitness trackers and smart scales to track additional metrics.

Importance: Medium

* Functional Requirement:

Gamification Elements:

Description: Introduce badges, achievements, and rewards to incentivize user engagement.

Importance: Medium

* Non-Functional Requirement:   
    
  Accessibility:

Description: Ensure the app is accessible to users with disabilities.

Importance: High

## **Document Outline**

The System Specification document is structured with key sections. The Executive Summary, on page 2, offers a brief overview of the FitFluencer app project, outlining objectives and next steps. The Introduction, on page 3, covers the problem statement, vision, and system capabilities. Moving to page 5, the Structural Model section includes class diagrams and metadata. Architecture Design, detailed on page 6, outlines the app's architecture and security plan. The User-Interface section, located on page 7, elucidates UI requirements and design. Lastly, page 8's Appendices contain a glossary, references, and supporting documentation. This approach ensures stakeholders efficiently understand FitFluencer app specifications and requirements.

# **Structural Model**

## **Model Introduction**

This section provides a detailed structural model of the system, designed to give developers a comprehensive understanding of the system's architecture through class diagrams and metadata. The structural model consists of two primary subsections.

## **Class Diagrams**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screen shot of a computer

Description automatically generated

## **Metadata**

* **User Class**

Description:

Represents a user in the FitFluencer app. Contains user credentials and methods for registration, login, and profile updates.

Attributes:

userID (int): Unique identifier for the user.

username (String): The user's username.

email (String): The user's email address.

password (String): The user's password.

dateOfBirth (Date): The user's date of birth.

Methods:

registerUser(email: String, password: String): boolean

Registers a new user.

loginUser(email: String, password: String): boolean

Logs in a user.

updateProfile(details: UserProfile): void

Updates the user's profile.

Pseudocode:

*registerUser(email, password):*

*if email and password are valid:*

*save user to database*

*return true*

*else:*

*return false*

*loginUser(email, password):*

*if email and password match records:*

*return true*

*else:*

*return false*

*updateProfile(details):*

*update user profile with new details in database*

* **UserProfile Class**

Description:

Represents a user profile, including bio, profile picture, and social media links.

Attributes:

profileID (int): Unique identifier for the profile.

userID (int): Identifier linking the profile to a user.

bio (String): User's bio.

profilePicture (String): URL to the profile picture.

socialMediaLinks (List<String>): List of social media links.

Methods:

editProfile(bio: String, profilePicture: String, socialMediaLinks: List<String>): void

Edits the user's profile details.

Pseudocode:

*editProfile(bio, profilePicture, socialMediaLinks):*

*update bio, profilePicture, and socialMediaLinks in database*

*FitnessClass Class*

*Description:*

*Represents a fitness class available for subscription.*

* **Subscription Class**

Description:

Represents a user's subscription to a fitness class.

Attributes:

subscriptionID (int): Unique identifier for the subscription.

userID (int): Identifier linking the subscription to a user.

fitnessClassID (int): Identifier linking the subscription to a fitness class.

startDate (Date): Subscription start date.

endDate (Date): Subscription end date.

Methods:

subscribe(userID: int, fitnessClassID: int): boolean

Subscribes a user to a fitness class.

cancelSubscription(subscriptionID: int): boolean

Cancels a subscription.

updateSubscription(subscriptionID: int, endDate: Date): boolean

Updates the subscription end date.

Pseudocode:

*subscribe(userID, fitnessClassID):*

*if userID and fitnessClassID are valid:*

*create subscription in database*

*return true*

*else:*

*return false*

*cancelSubscription(subscriptionID):*

*if subscriptionID exists:*

*delete subscription from database*

*return true*

*else:*

*return false*

*updateSubscription(subscriptionID, endDate):*

*if subscriptionID exists:*

*update endDate in database*

*return true*

*else:*

*return false*

# **Architecture Design**

## **Architecture Overview**

Section 4.1 offers a comprehensive view of the FitFluencer app's architectural design. It outlines the recommended structural approach, detailing key components, interactions, and deployment strategies. Technical readers will gain insights into implementing a 3-tier client-server architecture, leveraging cloud infrastructure for scalability and reliability, and potentially acquiring additional servers to bolster backend services and database management. This section serves as a detailed roadmap for understanding the app's technical underpinnings, guiding the development process towards a robust and scalable system.

## **Infrastructure Model**

* + 1. **Deployment Diagram 1 – Architecture Overview**

**A screenshot of a computer

Description automatically generated**

* + 1. **Deployment Diagram 2 – Nodes and Artifacts**

**A screenshot of a computer

Description automatically generated**

## **Hardware and Software Requirements**

* + 1. **Hardware Components**
* For the FitFluencer app's architecture, we'll need:

1. Web Servers: Hosting the web version.
2. Application Servers: For backend logic and APIs.
3. Database Servers: Storing user data.
4. Load Balancers: Distributing web and app traffic.

* Recommendation: Invest in new hardware for optimal performance. Users will need web browsers for the web version and iOS/Android devices for the respective applications.
  + 1. **Required Software Components**

1. Operating System: Linux (Ubuntu Server 20.04 LTS)
2. Database Management System: MySQL 8.0
3. Web Server Software: Apache HTTP Server 2.4
4. Application Server Software: Node.js 14.x
5. Programming Languages: JavaScript/TypeScript
6. Frameworks and Libraries: Express.js, React.js
7. Security Software: SSL/TLS Certificates, Firewall
8. Additional Software: Git, Docker
9. Development and Deployment Tools: IDE (e.g., Visual Studio Code), CI/CD Tools (e.g., Jenkins)

## **Security Plan**

* + 1. **Security Overview**

Potential security threats for the FitFluencer app include data breaches, injection attacks, authentication vulnerabilities, DoS attacks, and insufficient encryption. To mitigate these risks, implement SSL/TLS certificates and firewalls as required software components (section 4.3.2). SSL/TLS certificates ensure data confidentiality, while firewalls prevent unauthorized access and DoS attacks.

* + 1. **Security Plan**

|  |  |  |
| --- | --- | --- |
| **Threat Category** | **Potential Threats** | **Controls** |
| Physical Security | Theft or damage to server hardware | Implement physical access controls and surveillance |
| Network Security | Unauthorized access to user data during transmission | Encrypt data transmission using SSL/TLS protocols |
| Application Security | SQL injection attacks | Implement input validation and parameterized queries |
| File Security | Unauthorized access to stored user data | Implement file encryption and access control measures |
| User Security | Weak user passwords | Enforce password complexity requirements and 2FA |

# **User-Interface**

## **User-Interface Requirements and Constraints**

The user interface (UI) of the FitFluencer app is crafted for simplicity and ease of use, prioritizing a seamless user experience. Our guiding principles are ease of access, clear navigation, and visual appeal, while our main constraint is maintaining a clean layout without overwhelming the user.

* Home Page

The Home Page serves as the central hub for users to interact with the FitFluencer feed, showcasing posts from fitness influencers. Each post includes engagement buttons for liking, commenting, and sharing, making it easy for users to interact with the content. Below each post, users can find descriptions and comments, fostering a community where discussions can thrive. Navigation buttons at the bottom of the screen allow quick access to the homepage/feed and other sections of the app, ensuring users can easily stay updated and engaged.

* Login Page

The Login Page is designed for simplicity and efficiency. Users can quickly log in by entering their username and password into clearly labeled fields, with a prominent "Enter" button to proceed. New users have the option to sign up, making it easy to join the FitFluencer community.

In summary, the UI of the FitFluencer app focuses on providing a resourceful and engaging experience. The clear layout and intuitive navigation ensure users can easily access and interact with content, making their fitness journey both enjoyable and motivating.

## **Window/Screen Navigation Diagram &**

## **UI Wireframes**

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

## **Reports: "Formal Output" Design**

N/A

# **Appendices**

## **Glossary**

1. Activity Stream: A chronological feed displaying user and influencer updates and posts within the app.
2. Agile Methodologies: Project management frameworks emphasizing iterative development, collaboration, and flexibility.
3. Application Server: A server dedicated to running backend services and logic for an application.
4. Authentication: The process of verifying the identity of a user or system.
5. CI/CD Tools: Continuous Integration/Continuous Deployment tools used to automate code integration, testing, and deployment.
6. Class Diagram: A type of static structure diagram in UML that describes the structure of a system by showing its classes, attributes, methods, and relationships.
7. Cloud Infrastructure: A virtualized IT environment provided over the internet, enabling scalable and flexible computing resources.
8. Compliance: Adhering to laws, regulations, and guidelines relevant to the app's operations and data handling.
9. Content Moderation: Tools and policies used to monitor and manage user-generated content, ensuring it adheres to platform standards.
10. Data Encryption: The process of converting data into a coded format to prevent unauthorized access.
11. Deployment Diagram: A UML diagram that shows the physical arrangement of hardware and software components in a system.
12. Direct Messaging: A feature allowing users to communicate privately with others within the app.
13. Docker: A platform for developing, shipping, and running applications using containerization.
14. Functional Requirements: Specifications of what the system should do, describing the system’s functionality.
15. Gamification Elements: Features like badges, achievements, and rewards used to enhance user engagement through game-like elements.
16. Load Balancer: A device or software that distributes network or application traffic across multiple servers.
17. Macro Tracking: Monitoring the intake of macronutrients (proteins, fats, and carbohydrates) as part of fitness and diet management.
18. Metadata: Data that provides information about other data, such as descriptions of classes, attributes, and methods in a system.
19. Non-functional Requirements: Specifications describing the system's operation, including performance, security, usability, and reliability.
20. Parameterized Queries: Database queries that use parameters to dynamically insert values, helping prevent SQL injection attacks.
21. Scalability: The ability of a system to handle growing amounts of work or to be expanded to accommodate growth.
22. Security Plan: A comprehensive strategy outlining measures to protect the system from various security threats.
23. SSL/TLS Certificates: Digital certificates used to secure data transmission over the internet through encryption protocols.
24. Stakeholders: Individuals or groups with an interest in the development and success of the app, including users, developers, and investors.
25. System Architecture: The conceptual model defining the structure, behavior, and views of a system.
26. UML (Unified Modeling Language): A standardized modeling language used to visualize the design of a system.
27. Use Case: A description of a system’s behavior in response to external requests, detailing how users interact with the system.
28. User Interface (UI): The space where interactions between humans and machines occur, encompassing the design and layout of screens and controls.
29. Version Control: The management of changes to documents, programs, and other information stored as computer files.

## **References / Bibliography**

1. Lucidchart. (n.d.). Online Diagram Software & Visual Solution. Retrieved from [www.lucidchart.com]
2. Cameron, A. (2024). CSC 3150 Lecture Notes. Seattle Pacific University.
3. Johnson, M. (2022). The Rise of Fitness Influencers: Trends and Impact. Journal of Social Media Studies, 12(3), 45-67.
4. Smith, A. (2020). Social Media Marketing Strategies for Fitness Brands. New York: HarperCollins

## **Supporting documentation**

N/A