1. Introduction

1.1 Purpose

The purpose of this SRS is to define the requirements for the development of the Check-In software. It aims to provide a detailed understanding of the functionality, features, and constraints of the software.

1.2 Scope

The Check-In software will enable users to track their macros and progress photos, facilitating their fitness goals. It will include features such as account creation, user authorization and authentication, profile management, photo upload and comparison, input and tracking of fitness and nutrition metrics, and visualization of results through dynamic graphs.

1.3 Definitions, Acronyms, and Abbreviations

Provide a glossary of relevant terms, acronyms, and abbreviations used throughout the document.

1.4 References

List any references used in the creation of this document, such as existing system documentation, industry standards, or regulatory requirements.

2. User Requirements

2.1 Functional Requirements

2.1.1 Account Management

- The system shall allow users to create a new account by providing their name, email, and password.

- The system shall provide secure login functionality for users to access their accounts.

2.1.2 Profile Management

- Users shall be able to edit their profile information, including name, email, and profile picture.

2.1.3 Photo Management

- Users shall be able to upload progress photos and save them for future comparison.

- The system shall support the display and comparison of previously uploaded photos.

2.1.4 Metrics Tracking

- Users shall be able to input and track their weekly averages for weight, carbohydrates, proteins, and fats.

- The system shall provide a mechanism for users to view and analyze their tracked metrics.

2.1.5 Graphical Visualization

- The system shall generate graphical representations of the user's tracked metrics for easy visualization and analysis.

2.2 Non-Functional Requirements

2.2.1 Performance

- The system shall respond to user interactions within 2 seconds, providing a seamless user experience.

- The system shall support at least 100 concurrent users without significant performance degradation.

- The system shall have a minimum uptime of 99% to ensure availability.

2.2.2 Security

- User data shall be encrypted during transmission to protect against unauthorized access.

- The system shall implement secure user authentication mechanisms to prevent unauthorized access.

- Role-based access control shall be enforced to restrict user access based on assigned roles.

2.2.3 Usability

- The user interface shall be intuitive, user-friendly, and responsive across different devices and screen sizes.

- The system shall provide appropriate error messages and feedback to guide users in their interactions.

2.2.4 Reliability

- The system shall handle errors and exceptions gracefully, minimizing downtime and data loss.

- Data backups shall be performed regularly to ensure data integrity and availability.

2.2.5 Scalability

- The system architecture shall be designed to accommodate future growth and increasing user demands.

3. System Requirements

3.1 User Roles and Permissions

- Coaches shall have administrative roles to view client pages, while clients' pages shall be hidden from other users.

- Clients shall have the ability to authorize specific coaches to view their progress.

3.2 Data Management

The system shall use a relational database management system (e.g., MySQL) to store user data securely.

The database shall maintain appropriate data structures and relationships to ensure efficient data storage and retrieval.

The following data fields and structures shall be implemented:

User Table:

* User ID (Primary Key)
* Name
* Email
* Password (hashed and salted)
* Profile Picture

Progress Photo Table:

* Photo ID (Primary Key)
* User ID (Foreign Key referencing User Table)
* Photo File
* Timestamp

Metric Tracking Table:

* Metric ID (Primary Key)
* User ID (Foreign Key referencing User Table)
* Weight
* Carbohydrates
* Proteins
* Fats
* Timestamp

4. User Interface Design

The user interface shall adhere to modern design principles, with a clean and intuitive layout.

Branding elements, such as colors and logos, shall be incorporated into the design.

The interface shall be responsive, ensuring optimal user experience across different devices and screen sizes.

Navigation menus and controls shall be logically organized for ease of use.

User input forms shall provide clear labels and validation feedback.

5. Integration Requirements

The system shall integrate with external systems and APIs for functionalities such as:

Email notifications for account registration and password reset using a service like SendGrid or SMTP.

Cloud storage for storing uploaded photos, such as Amazon S3 or Google Cloud Storage.

Third-party authentication services (e.g., OAuth) for seamless login options.

Integration with these systems shall follow their respective documentation and guidelines.

6. Testing and Quality Assurance

A comprehensive testing strategy shall be implemented to ensure the quality and reliability of the software.

Test cases shall be designed and executed to verify both functional and non-functional requirements.

Automated testing frameworks, such as pytest or Selenium, shall be utilized for efficient and repeatable testing.

Regular code reviews shall be conducted to identify and address any code quality or design issues.

Continuous integration and deployment (CI/CD) practices shall be implemented to facilitate the testing and deployment process.

7. Project Constraints

The project budget shall be allocated for development, testing, deployment, and maintenance activities.

The project timeline shall be adhered to, with milestones and deliverables clearly defined.

Technical constraints, such as compatibility with specific programming languages, frameworks, or hosting environments, shall be considered during development.

Any regulatory or compliance requirements, such as data privacy or security regulations, shall be adhered to throughout the development process.