# FitTrack需求文档

## 1. 文档信息

文档标题：FitTrack健身跟踪app需求分析

版本：1.0

编写日期：2024/04/26

## 2. 项目概述

项目名称：FitTrack健身跟踪app

项目目标：FitTrack 是一款综合健身跟踪应用，旨在帮助用户记录、分析和管理他们的健身活动、饮食习惯和身体指标，从而实现健康生活目标。用户可以个性化设置目标、跟踪进度，并获得个性化建议和报告。

背景和目的：在当今社会，随着人们生活节奏的加快和工作压力的增大，越来越多的人开始意识到身体健康的重要性。健身活动、合理饮食以及身体指标的监测，已成为许多人追求健康生活的关键要素。然而，传统的健身方式往往缺乏系统性、科学性和可持续性，难以有效帮助用户达到健身目标。因此，开发一款能够综合记录、分析和管理健身活动的应用显得尤为迫切。

## 3. 范围定义

### 3.1项目范围

FitTrack应用开发项目的范围涵盖了健身跟踪应用的全面设计与开发，包括：  
 用户界面设计：设计简洁直观的用户界面，确保用户能够方便快捷地进行操作，包括登录、注册、目标设置、数据录入、进度查看等功能界面。

后端开发与服务器搭建：构建稳定可靠的后端服务器，实现用户数据的存储、处理和传输，确保应用的正常运行和数据安全。

数据库设计与构建：设计合理的数据库结构，存储用户的个人信息、健身目标、运动数据、饮食记录以及身体指标等关键信息，支持高效的数据查询和分析。

### 3.2功能实现

开发并优化核心功能，如个性化目标设置、实时数据跟踪、进度分析与报告生成、个性化建议推送等，以满足用户多样化的健身需求。

项目不包含的内容主要是一些非核心功能或与健身跟踪不直接相关的业务，例如社交娱乐功能、电商购物功能等。这些功能虽然可能提升用户体验，但并非FitTrack应用的主要定位和目标，因此不在本项目的开发范围内。

## 4. 需求详情

### 4.1 功能需求

#### 4.1.1 登录/注册页面

登录功能

登录方式：

手机号+短信验证码（SMS）

用户名+密码

调用微信API实现快速登录

自动登录：若设备过去有登录记录，下次打开应用时自动登录，无需重复输入登录信息。

人机验证：使用JWT（JSON Web Tokens）进行用户身份验证，确保登录过程的安全性。

注册功能

注册方式：

调用微信API，若用户已有微信用户名，则直接使用该用户名进行注册。

使用手机号注册，系统将为用户生成一个随机用户名。

完善用户信息：在注册过程中，用户需要完善以下信息：年龄、体重、性别、健身目标。系统根据用户信息提供兴趣推荐，并据此调整用户界面，以提供更加个性化的体验。

#### 4.1.2 活动功能

运动类型选择：用户可以从预定义的运动类型列表中选择自己所进行的活动。

时长记录：用户输入或使用应用内的计时器功能记录运动时长。

卡路里计算：根据用户选择的运动类型和时长，系统自动计算并显示消耗的卡路里。

#### 4.1.3 记录饮食摄入功能

食物输入：用户可以通过手动输入或选择预定义的食物来记录饮食摄入。

食物类型与卡路里计算：根据用户输入的食物类型，系统自动计算并显示摄入的卡路里。

#### 4.1.4 身体指标跟踪功能

指标输入：用户手动输入体重、体脂率、心率等身体指标。

历史记录查看：用户可以查看过去记录的身体指标数据，以便跟踪变化。

#### 4.1.5 目标设定与跟踪功能

目标公里数设定：用户可以设定自己的跑步目标公里数。

微信步数同步：应用可以与微信步数同步，实时更新用户的步数数据。

每日提醒：根据用户设定的目标，应用提供每日提醒功能，鼓励用户坚持跑步。

#### 4.1.6 数据可视化分析功能

调用百度API：应用调用百度API进行数据可视化分析，将用户的健身数据以图表形式展示，帮助用户更直观地了解自己的健身进展。

#### 4.1.7 智能建议功能（Bonus）

调用API：应用通过调用相关API，根据用户的健身数据、目标和偏好，提供个性化的健身建议和计划，帮助用户更有效地达到健身目标。

### 4.2 非功能需求

#### 4.2.1性能要求

响应时间：

用户界面的响应时间应控制在2秒以内，确保用户操作的流畅性。

数据同步和上传的响应时间应根据网络状况进行优化，确保在网络良好的情况下，数据同步时间不超过5秒。

处理能力：

应用应能够同时处理大量用户的并发请求，保证在高并发场景下系统运行的稳定性。

应用后端服务器应具备足够的数据处理能力，支持实时数据分析和报告生成。

#### 4.2.2可靠性和稳定性

故障率：

系统的故障率应控制在极低的水平，确保用户能够持续稳定地使用应用。

定期进行系统维护和升级，预防潜在故障的发生。

恢复机制：

在系统出现故障时，应能够快速定位问题并进行修复，确保服务的及时恢复。

建立数据备份和恢复机制，防止数据丢失，确保用户数据的安全性。

#### 4.2.3安全性

数据保护：

用户数据应进行加密存储和传输，防止数据泄露。

严格遵守数据保护法规，确保用户隐私不被侵犯。

用户授权：

实行严格的用户认证和授权机制，确保只有经过认证的用户才能访问系统。

对不同用户设置不同的权限级别，实现数据的分级访问和管理。

### 4.3 用户界面和用户体验

界面要求：界面布局、风格指南等。

用户体验目标：用户使用流程、易用性等。