

健康管理アプリ仕様書

24G1027 江戸颯沔

2025 年 1 月 7 日

1 はじめに

本仕様書は、体重管理アプリケーションの開発に関する仕様を記述したものである。本アプリケーションは、ユーザーが自分の体重、摂取した食べ物のカロリー、運動時間を記録し、それらのデータを元にグラフ化する機能を提供する。さらに、目標体重に対する進捗を表示する機能も含まれており、ユーザーが自分の健康管理をより効果的に行えるよう支援する。本仕様書では、利用者、管理者、開発者それぞれの視点から必要な機能とシステムの動作を詳述する。

2 利用者向け仕様

2.1 データ入力

利用者は、以下のデータを入力することができる。

- 体重 (kg)
- 身長 (cm)
- 摂取した食べ物 (カンマ区切り)
- 運動時間 (分)
- 目標体重 (kg) (オプション)

これらのデータを入力し、送信することで、体重やカロリー摂取、運動の記録を保存できる。入力したデータはその場で検証され、形式が正しい場合にのみ保存される。

2.2 データの視覚化

ユーザーが入力したデータは、折れ線グラフとして可視化され、体重やカロリー摂取の変動を一目で確認できる。以下の例は、体重の推移を示す折れ線グラフである。

健康管理アプリ

体重 (kg):

35

身長 (cm):

158

食べたもの (例: りんご, バナナ, 鶏むね肉, ご飯):

バナナ

運動時間 (分):

60

目標体重 (kg):

35

データを保存

保存されたデータ

日付: 2025-01-07, 体重: 44 kg, 摂取カロリー: 100 kcal, 運動時間: 0 分
日付: 2025-01-07, 体重: 45 kg, 摂取カロリー: 90 kcal, 運動時間: 0 分
日付: 2025-01-07, 体重: 43 kg, 摂取カロリー: 90 kcal, 運動時間: 20 分
日付: 2025-01-07, 体重: 45 kg, 摂取カロリー: 150 kcal, 運動時間: 40 分
日付: 2025-01-07, 体重: 46 kg, 摂取カロリー: 250 kcal, 運動時間: 30 分
日付: 2025-01-07, 体重: 44 kg, 摂取カロリー: 100 kcal, 運動時間: 30 分
日付: 2025-01-07, 体重: 38 kg, 摂取カロリー: 100 kcal, 運動時間: 60 分
日付: 2025-01-07, 体重: 40 kg, 摂取カロリー: 90 kcal, 運動時間: 60 分
日付: 2025-01-07, 体重: 36 kg, 摂取カロリー: 90 kcal, 運動時間: 60 分
日付: 2025-01-07, 体重: 35 kg, 摂取カロリー: 90 kcal, 運動時間: 60 分

現状と目標体重との差

目標体重に達しました！

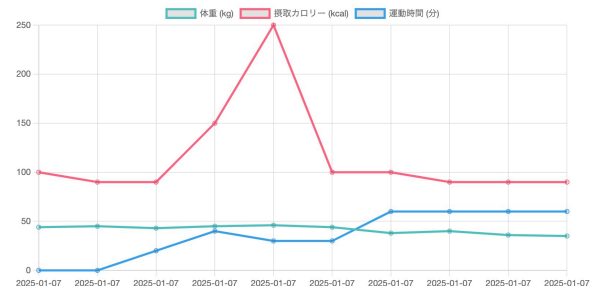


図 2 実際に体重やカロリー摂取の変動を示す折れ線グラフ

図 1 質問項目の例

ユーザーの目標体重との差が表示され、目標達成までの進捗を知ることができる。差がプラスの場合、目標体重に近づいていることを示し、マイナスの場合は目標体重をオーバーしていることを示す。進捗バーやアラート機能を搭載し、目標達成に向けてユーザーをサポートする。

3 管理者向け仕様

3.1 データの保存

管理者は、ユーザーが入力したデータを保存し、後でアクセスできるように管理する。データはサーバーに保存され、必要に応じて取り出し、表示することができる。データは日付ごとにグループ化され、アクセス効率が低い形式で保存される。

3.2 データベースの構造

データベースには以下の情報が保存される。

- 日付 (yyyy-mm-dd 形式)
- 体重 (kg)
- 摂取カロリー (kcal)
- 運動時間 (分)
- ユーザー ID (ユーザーごとに一意)

これらの情報を利用して、ユーザーごとの健康データを管理する。データベースは、複数のユーザー情報を効率的に取り扱えるよう、正規化されている。

3.3 バックアップ

定期的にデータベースのバックアップを取ることが推奨される。バックアップは安全な場所に保存し、データが失われないようにする。バックアップのスケジュールは自動化される。

4 開発者向け仕様

4.1 システム構成

本システムは以下の技術で構成されている。

- フロントエンド: HTML, CSS, JavaScript (Chart.js)
- バックエンド: Node.js (Express)
- データベース: 任意の RDB (例: MySQL, PostgreSQL)

フロントエンドは、ユーザーインターフェースを提供し、データの入力とグラフの描画を行う。バックエンドは、API の設計とデータ保存機能を提供し、データベースと連携する。

4.2 API 設計

データの保存は POST メソッドを使用して行い、サーバーとの双方向通信を行う。以下は、データを保存するための API エンドポイントの設計例である。

4.2.1 POST /save-data

データをサーバーに送信し、保存するエンドポイント。

- リクエストボディ: 表形式でのリクエスト例

4.2.2 リクエスト例

キー	値
date	2025-01-07
weight	70.5
calories	2500
exerciseTime	30

4.2.3 レスポンス例

図 3, 図 4 のようにレスポンスが行われる。



図 3 失敗例

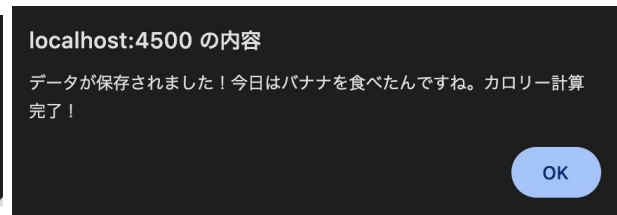


図 4 成功例

4.3 グラフ描画

本アプリケーションでは、ユーザーの体重、摂取カロリー、運動時間を視覚的に表示するために、グラフを利用する。グラフは、入力されたデータをもとに動的に生成され、ユーザーが自分の健康管理状況を一目で確認できるようにする。

グラフの描画には、JavaScript のライブラリである**Chart.js**を使用している。Chart.js は、簡単に折れ線グラフや棒グラフを描画できるため、データの推移を表示するのに適している。

- 体重の推移を表示するための折れ線グラフ
- 摂取カロリーの変動を示す棒グラフ
- 目標体重に対する進捗を示す進捗バー

以下は、Chart.js を使用して体重の推移を表示するためのコード例である。

```
var ctx = document.getElementById('weightChart').getContext('2d');
var weightChart = new Chart(ctx, {
  type: 'line', // グラフの種類（折れ線グラフ）
  data: {
    labels: ['2025-01-01', '2025-01-02', '2025-01-03', '2025-01-04'],
    // 日付
    datasets: [{
      label: '体重 (kg)',
      data: [70.5, 70.3, 70.0, 69.8], // 体重データ
      borderColor: 'rgba(75, 192, 192, 1)', // 線の色
      borderWidth: 2,
      fill: false // 塗りつぶしなし
    }]
  },
  options: {
    responsive: true,
    scales: {
      x: {
```

```

        type: 'category',
        title: {
            display: true,
            text: '日付'
        }
    },
    y: {
        title: {
            display: true,
            text: '体重 (kg)'
        }
    }
}
});

```

このコードでは、ユーザーが入力した体重データを折れ線グラフとして表示する。‘labels’には日付が、‘data’には対応する体重の値が設定される。グラフの外観は、‘borderColor’で線の色、‘borderWidth’で線の太さを指定している。

ユーザーインターフェースには、グラフを描画するための HTML のキャンバス要素を設置する。以下は、HTML のキャンバス要素の例である。

```
<canvas id="weightChart" width="400" height="200"></canvas>
```

このキャンバス要素に、JavaScript で設定したグラフを描画することができる。

4.4 採用技術の概要と理由

本アプリケーションでは、以下の技術を採用している。

- **Node.js (Express)**: 高速なサーバーサイド開発を行うために採用した。非同期処理が得意であり、高いパフォーマンスを発揮する。
- **Chart.js**: グラフ描画ライブラリ。視覚的にわかりやすいグラフを簡単に作成できるため、体重やカロリー摂取の推移をグラフ化するために使用している。
- **JSON**: データのやり取りには JSON 形式を採用した。シンプルで軽量であり、JavaScript との親和性が高いため。

5 GitHub リポジトリ

ソースコードは以下の GitHub リポジトリに置かれている。

- URL: <https://github.com/yourusername/your-repository>