Fullstack Test

Test Explaination

Test

1. Please summarize DB table required for System, response JSON data of the required API, use case, system configuration, work time, estimation, etc. in a diagram so that the connection of each is easy to understand.

2.Please develop Backend of the system ("Homepage" or "Your Record")
Please use one of the following Programming languages.

・Go
・JavaScript/TypeScript
・PHP

3.Please develop Frontend of the page developed in system 2 with React.

4.Please deploy so that the demo can work.

Addition:
- If you can't finish the implementation in time, decide on the scope and write the estimate for the undeveloped parts. Even if there is a part that is not implemented, please make sure it works.
- For a design overview, please refer to【Application Specification/ Overview】

Test Time

Please follow these guidelines:

- Estimated Testing time: 12 hours
- Maximum Testing time: 24 hours

Submission

Submit a archived zip of the Git repository folder

Please observe the following when zip archive
● Include .git folder in zip
● Delete any files that are not under git control

※Do not share it as public repository.

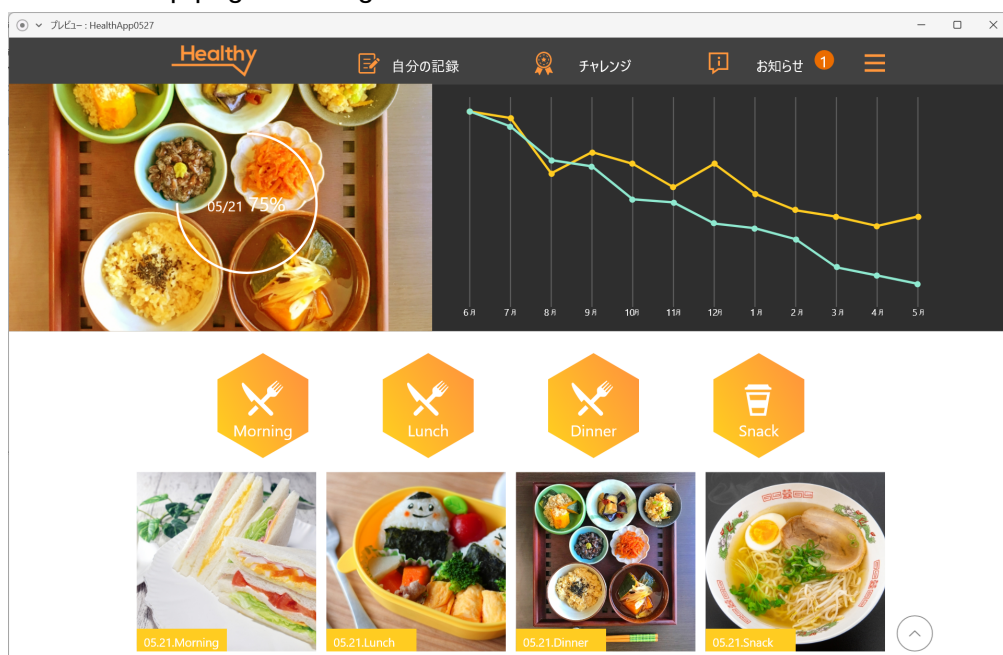# Application Specification/ Overview

A web application for health management

The design is in the figma here:
[HealthApp_1203 – Figma](#)

Top page

This is the top page after login.

The outline of each display is as following:

日付・達成率

Date/Achievement rate

体重・体脂肪率グラフ

Weight/Body Fat Percentage Graph

入力へ遷移するボタン

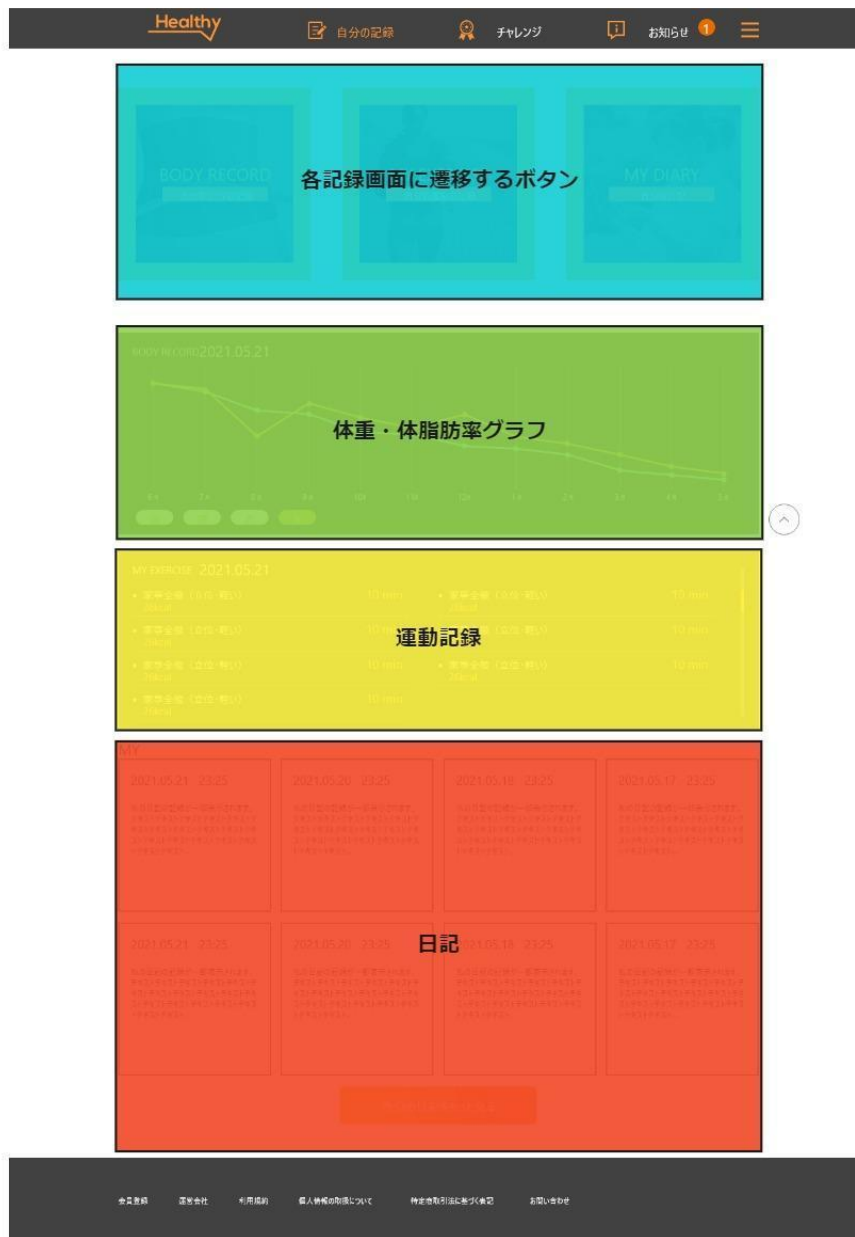 Button to transition to input

食事履歴

Meal history

My record

This page displays your own records.

The outline of each display is as following:

各記録画面に遷移するボタン
Button to transition to each recording screen

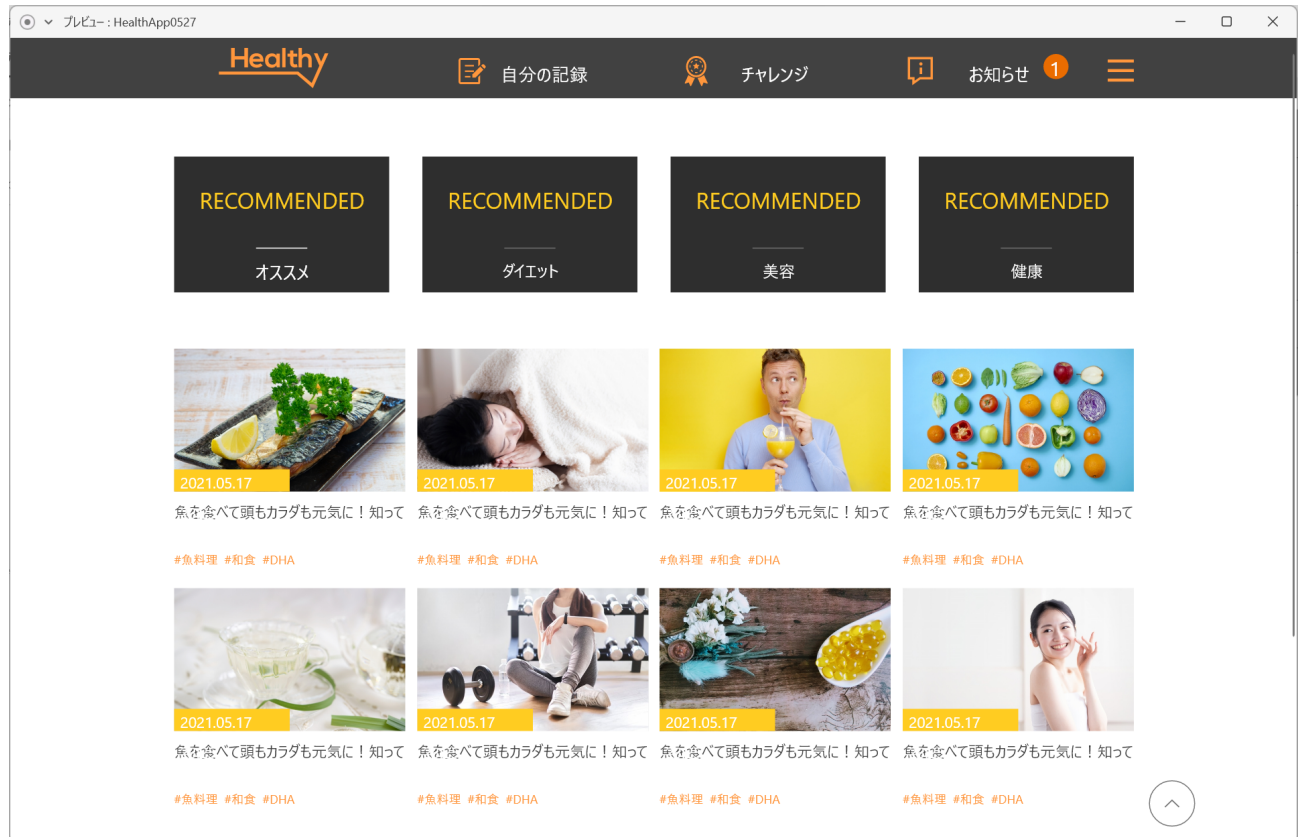体重・体脂肪率グラフ
Weight/Body Fat Percentage Graph

運動記録
Exercise record

日記
Diary

Column page

This is the page of the column about health. You can view it without logging in.

Evaluation criteria

We will review your submission by focusing on the following criteria:

- Clarity
  - Does the README clearly and concisely describe the problem, solution, and verification procedure?
  - Is the diagram easy to understand for other related developers such as front desk and infrastructure personnel?

- Code quality
  - Is the split appropriate?
  - Is the code simple, easy to understand, and easy to maintain?
  - Does your coding style match the language's guidelines?
  - Are you consistent in your codebase?