

Project Design Document

Project Name: HealthTracker

TABLE OF CONTENTS

1. System Overview
 - a. Brief Project Description
 - b. System Architecture
 - c. Technology Stack
2. Implementation Details
 - a. Codebase Structure
 - b. Key Implementations
 - c. Component Interfaces
3. Use Case Support in Design
 - a. Use Case Selection
 - b. Requirement Mapping
 - c. Use Case Design
4. Design Decisions
 - a. Technology Comparisons
 - b. Decision Justifications
5. GitHub Commit Requirement
 - a. Code Implementations & Interfaces

LIST OF CONTRIBUTORS

- Kerem ELMA
- Kaan ÖNEN
- Mehmet ESKİ

1. System Overview

a. Brief Project Description

HealthTracker is a comprehensive health tracking application designed to help users manage their daily health habits. It aims to centralize the tracking of important health activities, such as medication schedules, doctor appointments, fitness goals, and hydration levels, in one platform. The application is developed to improve user wellness by ensuring they never miss a crucial health task and promoting a proactive approach to maintaining overall health.

b. System Architecture

Since we are a development team of 4 people and are developing a small-scale project, it would be appropriate to use a monolithic structure for the project's general system architecture.

c. Technology Stack

The project we are developing will run on two different platforms. For the part that will run on the web platform, Python, HTML, and CSS will be used. For the backend, the Python framework Django will be used, and for the frontend, the JavaScript framework React.js will be used.

2. Implementation Details

a. Codebase Structure

Since the backend of the project's web development will be built on Django; router modules, form, model, and view modules that will interact with the pages will be located under the `backend/` path. Once the backend is completed, frontend development will begin under the `frontend/` path.

b. Key Implementations

Key implementations to be made:

- User login/register system
- Graphically represented statistics on the frontend
- Trackers such as water and appointments

c. Component Interfaces

Since there will be no database in the project, components will be needed on the backend side to process form data and return it in its final format. Similarly, communication between the backend and frontend will be established through these JSON files. Appropriate interfaces will also be included.

3. Use Case Support in Design

a. Use Case Selection

- i. **Medication Tracking and Reminder:** The user logs into the system, adds their medication with its name, dosage, and time. The system sends a notification when the specified conditions are met.
- ii. **Doctor Appointment Management:** The user logs into the system, enters the date, time, and doctor's name to add a new appointment. The appointment is saved, and a notification is sent when the time approaches. Appointments can be canceled or updated.
- iii. **Fitness Tracking:** The user logs into the system, can manually enter exercise information, and the system tracks whether daily goals are achieved. The user can review past activities in graphs.
- iv. **Hydration Reminder:** The user logs into the system, the system sets a daily water goal, and reminds the user to drink water at specified time intervals. The user enters the amount of water they have drunk into the system. The system shows whether the daily goal has been reached.

b. Requirement Mapping

First use case uses "Medication Tracking" requirement.

Second use case uses "Doctor Appointment Management" requirement.

Third use case uses "Fitness Tracking" requirement.

Fourth use case uses "Hydration Reminder" requirement.

c. Use Case Design

First of all, the login/register module will be used in all use cases.

After that, in first use case; medication adding, editing modules will be used, and then the module that will send a notification to the user when the necessary conditions are met will be activated.

In second use case, doctor appointment management module will be used, after that the module that will send a notification to the user when the necessary conditions are met will be activated. Also the user can be delete or update the appoint from the appointment management module.

In third use case, fitness and exercise management modules will be used for entering exercise information, also meanwhile the system track modules will be used. After that, in the frontend part, graph components will be represent the output.

In fourth use case, hydration reminder module will be used for tracking the amount of water has been consumed. Before all that, the module that calculates the liter daily water intake for the user. After calculation and according to the user input, graph components will be represent the output.

4. Design Decisions

a. Technology Comparisons

Due to the use of Django, the Python infrastructure will be used and we will be able to develop faster with less code. Unlike Express.js, Node.js-based projects, it works more automated, but it will be slower compared to a JavaScript-based language. Since we will be working with a small data set, processing files in JSON format rather than using a database will do the trick and also provide faster processing.

b. Decision Justifications

The reason Django was chosen for the backend is that some members of the developer team have experience in this area and it will facilitate the development process. The choice of React for the frontend is due to it being a popular frontend framework used today and our desire to integrate it into the project.

5. GitHub Commit Requirement

a. Code Implementations & Interfaces

Every newly added feature is kept in the GitHub repository. The documentation currently being prepared is also kept in their respective folders.

TASK MATRIX

Kerem ELMA	System Overview Implementation Details
Kaan ÖNEN	Use Case Support in Design Design Decisions
Mehmet ESKİ	GitHub Commit Requirement