FitTrack

TechOtakus: Joshua McArdle, Yovanny Moscoso, Khizar Saud, Seth Kelley, Cesar Rojas

1. Introduction

In today's digital era, many face health issues due to sedentary lifestyles but lack a single platform to track and manage their fitness. FitTrack offers a web app that combines real-time health tracking with community interaction and expert advice. The goal is to create a user-friendly tool that balances data insights with community support for a comprehensive approach to wellness.

2. Proposed System

FitTrack is a web application that serves as a pivotal tool in not only reminding and motivating individuals to stay active but also in providing valuable insights into their health journey. Upon creating an account, users will be able to access numerous functionalities such as setting their goals for weight loss/gain/maintenance, searching and logging food, searching and logging exercises, and interacting with an AI chatbot about health/fitness related queries. Users will also be able to see the caloric content for a given food and numerous details such as important macros (fat/protein/carbohydrate), and when selecting a certain exercise they will be able to see how to perform the exercise and understand what target muscles that particular exercise hits. In utilizing these functionalities, the system will allow users to easily track their calories and nutrition throughout the week and set themselves on the path towards a smooth fitness journey.

2.1 Functional Requirements

User Authentication

r1: User shall be able to sign up.

r1.1: System shall validate if the username already exists during sign-up.

r1.2: System shall store the new user's details in the database.
r2: User shall be able to log in.

r2.1: System shall validate the email and password during login.

r2.2: System shall set session variables for a logged-in user.

Dashboard Access

r3: User shall be able to access the dashboard only when logged in.

r3.1: System shall redirect unauthorized users to the login page.

r3.2: System shall display BMR, TDEE, and daily calorie limit if available.

Food Logging

r4: User shall be able to log food items.

r4.1: System shall fetch calorie count for a given food name via an API.

r4.2: System shall update the user's daily calorie intake in the database.

Goal Setting

r5: User shall be able to set fitness goals.

r5.1: System shall calculate BMR, TDEE, and daily calorie limit based on the user input.

r5.2: System shall store the goals in the database.

Communication

r6: User shall be able to send messages to the fitness assistant.

r6.1: System shall store the chat history in the database.

r6.2: System shall generate responses using OpenAI API.

r7: User shall be able to send emails through the contact form.

r7.1: System shall send a confirmation email to the user.

Location and Weather Services

r8: User shall be able to get the nearest gym based on their location.

r8.1: System shall fetch the nearest gym details using Google Places API.

r9: User shall be able to get the current weather based on their location.

r9.1: System shall fetch the current weather details using OpenWeatherMap API.

Logout

r10: User shall be able to log out.

r10.1: System shall clear the user's session variables on logout.

Password Recovering

r11: The user shall have access to a "Forgot My Password" feature to request password resets.

r12: The user shall be provided with a secure email link to users for initiating the password reset process.

r12.1: The system shall set a time limit for the validity of password reset links.

r12.2: The system shall verify the user's identity before allowing password reset.

Contact us

r13: The user shall be able to access contact form to send inquiries, feedback, or support requests.

r13.1 The system shall include fields for username, email, subject, message, and additional relevant information.

r13.2 The system shall validate user input, ensuring it adheres to required formats (e.g., valid email addresses).

Meal Recommendation:

r14: The user shall receive personalized meal recommendations based on their input and goals.

r14.1: The system shall provide meal recommendations based on the user goals.

r15: The user shall have the option to add a meal recommendation to their daily meal plan.

r15.1: The system shall provide a feature to add a meal to their daily meal plan and their macros to the user daily progress.

Exercise Logging

r16: The user shall be able to search from a list of available exercises.

r16.1: The user shall be able to select an exercise from the list and view its information.

r17: The user shall be able to log individual exercises.

2.2 Non-Functional Requirements

System Performance

nf1: The application shall run on a Raspberry Pi with 8GB RAM to ensure fast performance.

nf1.1: The system shall be able to handle at least 10 simultaneous users without a noticeable decrease in performance.

DNS & Hosting

nf2: The application shall use No-IP for DNS management.

nf2.1: The system shall ensure that domain name resolution via No-IP is reliable and has low latency.

Security

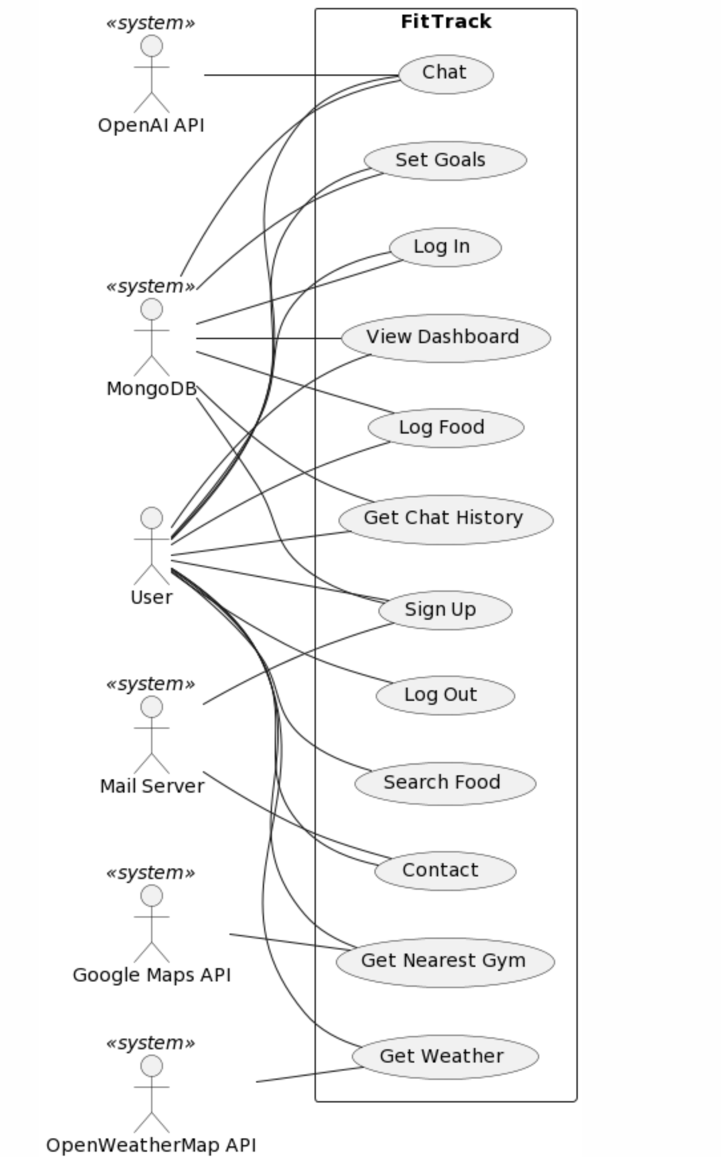
nf3: The application shall use TLS for secure data transmission.

nf3.1: All data transmitted between the client and server shall be encrypted using TLS.

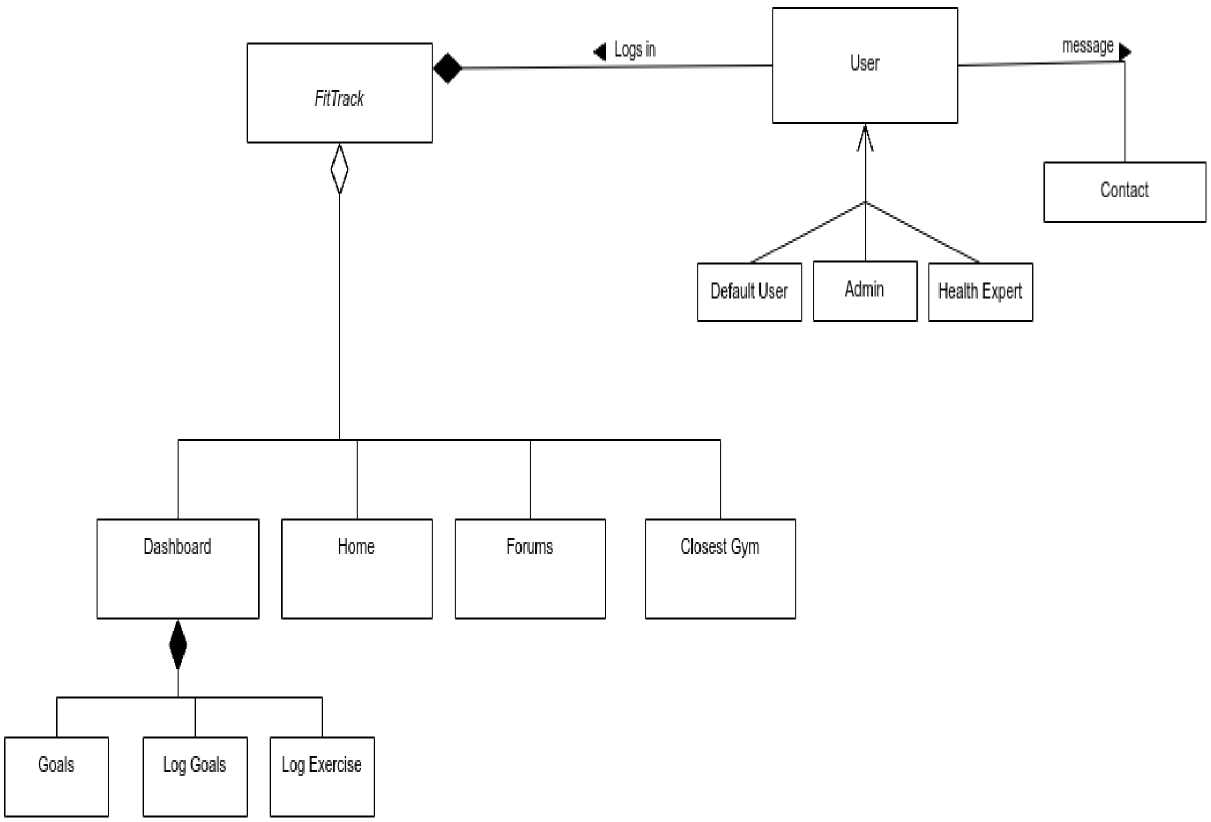
nf3.2: The system shall handle session management securely, preventing session hijacking or fixation attacks.

2.3 System Models

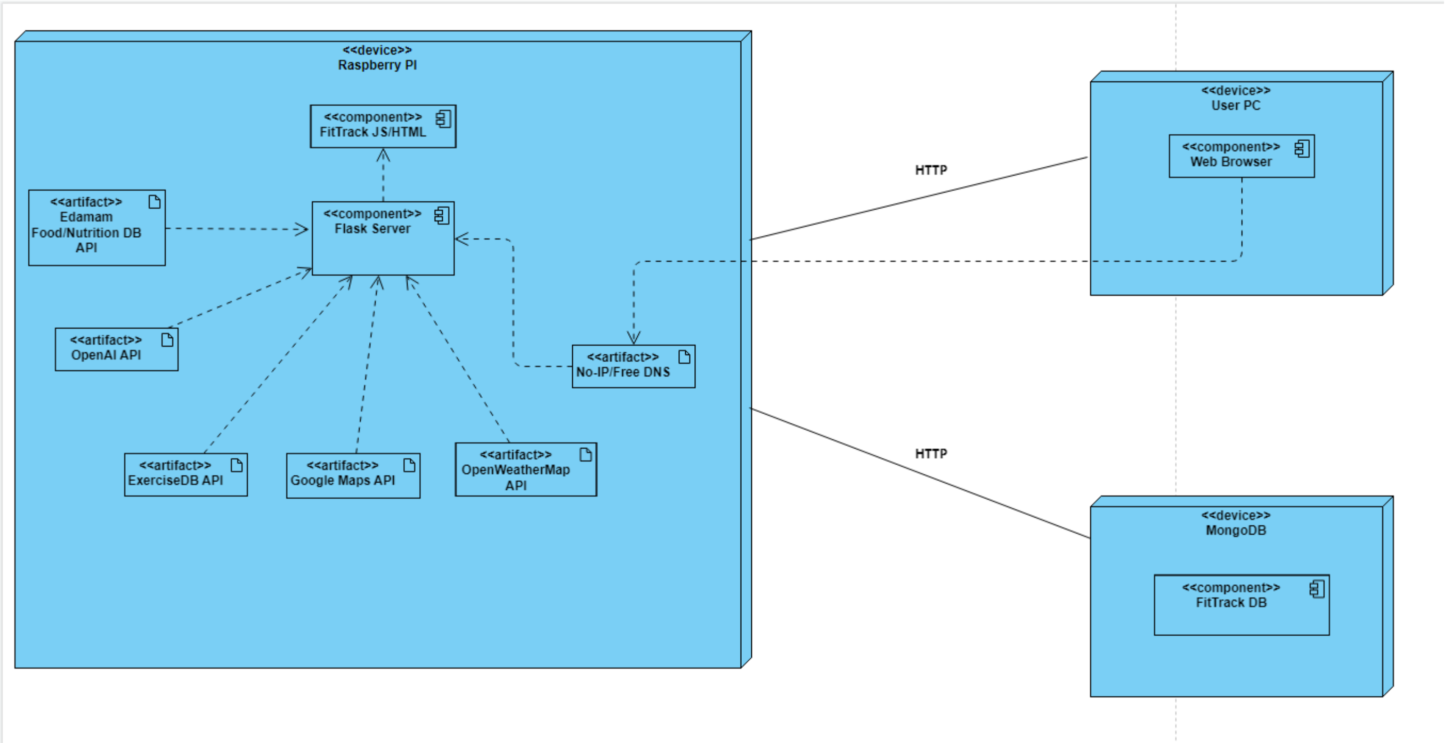
2.3.1 Use Case Diagram



2.3.2 Domain Model - Class Diagram



2.3.3 Deployment Diagram


<<device>>
Raspberry PI
<<component>>
FitTrack JS/HTML
<<artifact>>
Edamam Food/Nutrition DB API
<<component>>
Flask Server
<<artifact>>
OpenAI API
<<artifact>>
No-IP/Free DNS
<<artifact>>
ExerciseDB API
<<artifact>>
Google Maps API
<<artifact>>
OpenWeatherMap API
HTTP
<<device>>
User PC
<<component>>
Web Browser
HTTP
<<device>>
MongoDB
<<component>>
FitTrack DB