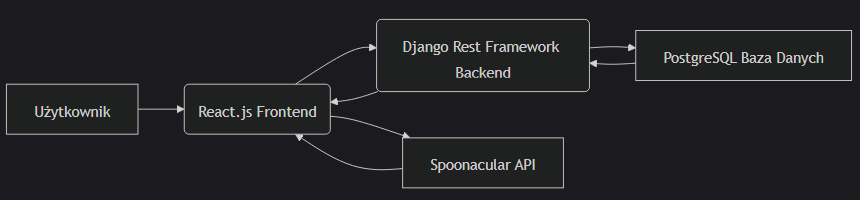
**Technical Design**

**1. Wprowadzenie**

Celem dokumentu jest przedstawienie technicznego projektu aplikacji żywieniowej, która działa w stylu Fitatu. Aplikacja ma na celu umożliwienie użytkownikom śledzenia ich diety, planowania posiłków oraz monitorowania postępów w odchudzaniu lub budowaniu masy mięśniowej.

**2. Architektura Aplikacji**

Aplikacja będzie zbudowana w architekturze klient-serwer, gdzie frontend będzie odpowiedzialny za interakcję z użytkownikiem, a backend za przetwarzanie danych i logikę biznesową.



**2.1. Frontend**

* Technologia: React.js
* Zadania:
* Tworzenie interfejsu użytkownika.
* Przesyłanie danych wejściowych do BackEndu
* Walidacja danych wejściowych użytkowników.
* Komunikacja z backendem za pomocą JWT.
* Pobieranie danych jedzenia z zewnętrznego API

**2.2. Backend**

* Technologia: Django Rest Framework
* Zadania:
* Obsługa logiki biznesowej.
* Zarządzanie bazą danych.
* Udostępnianie API dla frontendu.

**2.3. Zewnętrzne API**

Baza danych dużej częśći posiłków i produktów będzie pobierana z zewnętrznego API Spoonacular.

**3. Walidacja Użytkowników**

Walidacja użytkowników będzie realizowana na frontendzie oraz backendzie:

* Frontend: Przy poprawnym wypełnieniu formularza wysyła dane za pomocą fetch’a do BackEndu, gdzie ten dodaje potrawe, użytkownika lub loguje użytkownika i zwraca tokeny.
* Backend: Walidacja danych przy użyciu simple-jwt do autoryzacji użytkowników.

**4. Model Bazy Danych**

Baza danych będzie zawierała następujące modele:

**4.1. User:**

* id: Integer (Primary Key)
* username: String (Unique)
* password: String (Hashed)

**4.2. UserData**

* id: Integer (Primary Key)
* user\_id: Integer (Foreign Key do Użytkownika)
* BirthDate: Date
* Weight: Float
* Height: Float

**4.3. Dish**

* id: Integer (Primary Key)
* name: String
* calories: Float
* protein: Float
* carbs: Float
* fat: Float
* user\_id: Integer (Foreign Key do Użytkownika)

**4.4. Diet**

* id: Integer (Primary Key)
* date: Date
* meal\_id: Integer (Foreign Key do Posiłku)
* user\_id: Integer (Foreign Key do Użytkownika)

**5. Technologie**

* Frontend: React.js Axios (do komunikacji z API), Html, Css
* Backend: Django, Django Rest Framework, PostgreSQL (jako baza danych)
* Zewnętrzne API: Spoonacular API

**6. Podsumowanie**

Aplikacja żywieniowa w stylu Fitatu będzie wykorzystywać nowoczesne technologie, aby zapewnić użytkownikom intuicyjne i efektywne narzędzie do zarządzania dietą. Dzięki zastosowaniu React.js na frontendzie oraz Django Rest Framework na backendzie, aplikacja będzie wydajna i łatwa w utrzymaniu. Zewnętrzne API Spoonacular dostarczy bogaty zbiór przepisów, co wzbogaci doświadczenie użytkowników.