

# Lista Kontrolna Analizy i Integracji (Checklist)

Zgodnie z Państwa zleceniem, proces tworzenia ostatecznego dokumentu specyfikacji technicznej i merytorycznej dla aplikacji treningowej opartej na AI będzie przebiegał według następujących kroków:

- **Ekstrakcja aksjomatów fizjologicznych:** Przeprowadzę głęboką analizę obu dokumentów źródłowych oraz dodatkowych materiałów badawczych (ACSM, NSCA) w celu wyizolowania sztywnych reguł dotyczących objętości, intensywności i doboru ćwiczeń, odrzucając niespójne lub przestarzałe zalecenia.
- **Unifikacja modelu danych (JSON):** Połączę wymagania dotyczące struktury danych z obu plików – scalając proste definicje ćwiczeń z pierwszego pliku z zaawansowanymi parametrami (Tempo, RPE, Superserie) z drugiego – tworząc jeden, kompletny i walidowalny schemat JSON.
- **Rozstrzyganie konfliktów logicznych:** W miejscach, gdzie dokumenty różnią się podejściem (np. sztywne vs. zakresowe liczby powtórzeń), przyjmę rozwiązanie bardziej elastyczne i sprzyjające autoregulacji (zakresy), co jest zgodne z nowoczesną metodyką treningową.
- **Optymalizacja pod kątem AI i Prompt Engineering:** Zintegruję instrukcje dotyczące interakcji z modelem językowym (LLM), aby zapewnić deterministyczne i bezpieczne generowanie planów, minimalizując ryzyko halucynacji.
- **Weryfikacja kompletności:** Sprawdzę, czy końcowy dokument pokrywa wszystkie aspekty: od onboardingu medycznego, przez strukturę makrocyklu, aż po logikę progresji ("Live Tracking"), zapewniając gotowość do implementacji w kodzie.

## Architektura Systemów Generowania Adaptacyjnych Planów Treningowych: Synteza Protokółów Fizjologicznych i Inżynierii Danych

### 1. Wstęp i Filozofia Systemu

Projektowanie aplikacji fitness nowej generacji, która wykorzystuje sztuczną inteligencję do pełnienia roli wirtualnego trenera, wymaga radykalnego odejścia od statycznych baz danych na rzecz dynamicznych silników decyzyjnych. Niniejszy raport stanowi kompleksową specyfikację techniczną i merytoryczną, łączącą rygorystyczne standardy fizjologiczne American College of Sports Medicine (ACSM) oraz National Strength and Conditioning Association (NSCA) z nowoczesnymi paradygmatami inżynierii oprogramowania, takimi jak architektura "Feature-First" i ścisła serializacja danych JSON.

Podstawowym założeniem jest **bio-wierność obliczeniowa**. System nie może traktować planu treningowego jako luźnej listy zadań, lecz jako precyzyjnie skonstruowany obiekt danych, w którym zmienne takie jak intensywność, objętość (Volume Load), gęstość i częstotliwość są matematycznymi

determinantami specyficznych adaptacji neuro-mięśniowych.<sup>1</sup> Dokument ten integruje wymogi dotyczące zaangażowania użytkownika i czytelności interfejsu<sup>1</sup> z twardymi danymi naukowymi<sup>2</sup>, tworząc spójny ekosystem, w którym każda linijka kodu przekłada się na realny bodziec fizjologiczny.

## 2. Teoretyczne Ramy Komputacyjnego Projektowania Treningu

Aby algorytm AI mógł skutecznie generować plany treningowe, musi operować na fundamencie zasady specyficzności (SAID – *Specific Adaptation to Imposed Demands*). Oznacza to, że system musi posiadać zdolność tłumaczenia deklaracyjnych celów użytkownika (np. "chcę być silniejszy") na imperatywne parametry mechaniczne i metaboliczne. Poniższa sekcja definiuje logikę, którą silnik AI musi stosować przy doborze zmiennych treningowych.

### 2.1. Kontinuum Adaptacji Neuro-Mięśniowej

System decyzyjny musi rozróżniać trzy główne ścieżki adaptacyjne, z których każda wymaga odmiennej konfiguracji parametrów w generowanym obiekcie JSON.

#### 2.1.1. Siła Mięśniowa (Efektywność Neurologiczna)

Siła, definiowana jako zdolność układu nerwowo-mięśniowego do generowania maksymalnej siły zewnętrznej, jest cechą wymagającą optymalizacji pod kątem intensywności, a nie objętości metabolicznej.<sup>1</sup>

- **Mechanizm Fizjologiczny:** Adaptacja zachodzi głównie w układzie nerwowym poprzez zwiększenie rekrutacji jednostek motorycznych oraz częstotliwości wyładowań (rate coding). Proces ten jest wysoce energochłonny i opiera się na systemie fosfagenowym (ATP-PCr).
- **Parametry Algorytmiczne:**
  - **Obciążenie (Load):** Algorytm musi preskrybować obciążenia w zakresie 80–100% ciężaru maksymalnego (1RM) dla osób zaawansowanych oraz 60–80% dla początkujących, co jest niezbędne do rekrutacji wysokoprogowych jednostek motorycznych.<sup>2</sup>
  - **Powtórzenia (Reps):** Niska liczba, w zakresie 1–6 powtórzeń w serii.
  - **Czas Przerwy (Rest):** Jest to zmienna krytyczna, często błędnie implementowana w prostych aplikacjach. Resynteza ATP jest procesem czasochłonnym; około 50% zasobów odnawia się w 30 sekund, ale pełna regeneracja wymaga 3 do 5 minut. Algorytm generujący plan siłowy **musi** wymuszać przerwy rzędu 180–300 sekund (rest\_period\_seconds  $\geq$  180). Skrócenie tego czasu prowadzi do pracy w warunkach długu tlenowego, co zmienia charakter treningu z siłowego na wytrzymałościowy, niwecząc cel użytkownika.<sup>2</sup>

#### 2.1.2. Hipertrofia Mięśniowa (Adaptacja Strukturalna)

Jest to najczęstszy cel użytkowników aplikacji komercyjnych, polegający na zwiększeniu fizjologicznego przekroju poprzecznego mięśnia (CSA).

- **Mechanizm Fizjologiczny:** Proces ten jest napędzany przez trzy mechanizmy: napięcie mechaniczne, stres metaboliczny i uszkodzenia mięśniowe. Wymaga to balansu między intensywnością a całkowitą objętością pracy.

- **Parametry Algorytmiczne:**
  - **Obciążenie:** 70–85% 1RM.
  - **Objętość Całkowita (Volume Load):** Jest kluczowym wskaźnikiem ( $\text{Serie} \times \text{Powtórzenia} \times \text{Ciężar}$ ). Dla osób zaawansowanych system powinien generować od 3 do 6 serii roboczych na ćwiczenie, celując w 10–20 serii na grupę mięśniową w skali tygodnia.<sup>3</sup>
  - **Powtórzenia:** 6–12. Zakres ten zapewnia optymalny czas pod napięciem (TUT) dla akumulacji metabolitów (mleczanów, jonów  $H^+$ ), co stymuluje sygnalizację anaboliczną.
  - **Czas Przerwy:** 60–90 sekund. Niepełny wypoczynek jest tu pożądany, aby utrzymać wysoki poziom stresu metabolicznego.<sup>1</sup>

### 2.1.3. Wytrzymałość Mięśniowa (Wydajność Metaboliczna)

Zdolność do wykonywania pracy podmaksymalnej przez wydłużony czas.

- **Mechanizm Fizjologiczny:** Zwiększenie gęstości mitochondrialnej, kapilaryzacji oraz zdolności buforowania jonów wodorowych.
- **Parametry Algorytmiczne:**
  - **Obciążenie:** <70% 1RM.
  - **Powtórzenia:** >12 (często 15–25).
  - **Czas Przerwy:** Bardzo krótki (<60 sekund, często 30 sekund). Wymusza to na organizmie szybką utylizację produktów przemiany materii.<sup>1</sup>

## 2.2. Hierarchia i Sekwencjonowanie Ćwiczeń

Kolejność ćwiczeń w generowanym JSON nie może być losowa. Algorytm sortujący musi przestrzegać ściślejszej hierarchii opartej na wydatku energetycznym i bezpieczeństwie neurologicznym.<sup>2</sup>

1. **Prawo Priorytetu Złożoności (Compound First):** Ćwiczenia wielostawowe angażujące duże grupy mięśniowe (np. Przysiady, Martwy Ciąg, Wyciskanie Żołnierskie) muszą bezwzględnie znajdować się na początku bloku głównego ("Main Workout"). Angażują one najwięcej jednostek motorycznych i generują największe zmęczenie ośrodkowego układu nerwowego (CNS). Umieszczenie ich na końcu treningu, w stanie zmęczenia obwodowego, drastycznie zwiększa ryzyko błędu technicznego i kontuzji.
2. **Prawo Izolacji (Isolation Second):** Ćwiczenia jednostawowe (np. Uginanie ramion, Wyprosty nóg) powinny być planowane po ćwiczeniach głównych. Służą one do precyzyjnego dopracowania mięśni lub zwiększenia objętości bez nadmiernego obciążania układu nerwowego.
3. **Zasada Priorytetu Celu:** Badania wskazują, że ćwiczenia wykonywane na początku sesji przynoszą największe korzyści adaptacyjne. Jeśli w procesie onboardingu użytkownik zaznaczy "Priorytet: Klatka Piersiowa", algorytm powinien zmodyfikować standardową hierarchię, umieszczając Wyciskanie Leżąc jako pierwsze ćwiczenie, nawet w treningu typu Full Body.<sup>1</sup>

## 3. Segmentacja Użytkowników i Profilowanie (Onboarding AI)

Precyzyjne określenie punktu startowego użytkownika jest fundamentem bezpieczeństwa i skuteczności. Błędna klasyfikacja prowadzi do "zimnego startu", gdzie generowany plan jest albo zbyt łatwy (brak

postępów), albo zbyt trudny (ryzyko urazu). System odrzuca binarny podział na rzecz trójstopniowej macierzy kompetencji.<sup>1</sup>

### 3.1. Szczegółowa Charakterystyka Poziomów Zaawansowania

#### Poziom 1: Początkujący (Novice - Neurological Learner)

- **Kryterium:** Osoba z stażem <6 miesięcy ciągłego treningu lub powracająca po wieloletniej przerwie.
- **Cel Fizjologiczny:** Nauka wzorców ruchowych i adaptacja neurologiczna (koordynacja wewnątrz- i międzymięśniowa). Przyrost masy mięśniowej jest w tej fazie drugorzędny.
- **Konfiguracja Planu:**
  - **Struktura:** Trening całego ciała (FBW) 2–3 razy w tygodniu.
  - **Objętość:** Niska. 1–3 serie na ćwiczenie. Większa objętość u nowicjuszy jest klasyfikowana jako "śmieciowa objętość" (*junk volume*), która wydłuża regenerację nie przynosząc dodatkowych korzyści adaptacyjnych.<sup>3</sup>
  - **Dobór Ćwiczeń:** Wzorce ruchowe w stabilnych warunkach (np. Goblet Squat zamiast Back Squat, maszyny zamiast wolnych ciężarów przy niskiej ocenie mobilności).
  - **Tempo:** Wolne i kontrolowane (np. 3010 lub 4010) w celu wyrobienia kontroli motorycznej.

#### Poziom 2: Średniozaawansowany (Intermediate - Structural Builder)

- **Kryterium:** Od 6 miesięcy do 2 lat regularnego treningu. Adaptacje neurologiczne uległy spowolnieniu ("plateau newbie gains").
- **Cel Fizjologiczny:** Maksymalizacja hipertrofii i siły poprzez manipulację objętością.
- **Konfiguracja Planu:**
  - **Struktura:** Podział dzielony (Split), np. Góra/Dół (Upper/Lower) lub Push/Pull, 3–4 razy w tygodniu.
  - **Intensywność:** 60–80% 1RM.
  - **Periodyzacja:** Falowa (Undulating). Liniowa progresja (dodawanie ciężaru co trening) przestaje być efektywna; system musi wprowadzać zmienność obciążeń w skali tygodnia (np. Dzień Ciężki, Dzień Objętościowy).<sup>1</sup>

#### Poziom 3: Zaawansowany (Advanced - Performance Optimizer)

- **Kryterium:** Wieloletni staż, bliskość genetycznego potencjału siłowego.
- **Cel Fizjologiczny:** Przełamywanie homeostazy poprzez ekstremalne bodźce.
- **Konfiguracja Planu:**
  - **Struktura:** Wysoka częstotliwość (4–6 dni w tygodniu), często splity typu Push/Pull/Legs lub specyficzne bloki priorytetowe.
  - **Techniki Intensyfikacji:** Schemat JSON musi obsługiwać metody takie jak: Superserie, Drop-sets, Cluster Sets, Powtórzenia Negatywne.
  - **Polaryzacja:** Trening zawiera zarówno bardzo ciężkie serie (1–5 powtórzeń, >85% 1RM) jak i serie metaboliczne (>15 powtórzeń) w ramach jednego mezocyklu.<sup>1</sup>

### 3.2. Dynamiczna Walidacja w Procesie Onboardingu

Formularz statyczny jest niewystarczający. Aplikacja powinna implementować "Assessment Workout" – pierwszą sesję kalibracyjną. System musi zbierać dane krytyczne dla logiki generowania:

- **Ograniczenia Mobilności:** Np. test przysiadu – niemożność wykonania pełnego przysiadu bez odrywania pięt powinna skutkować wykluczeniem *Barbell Back Squat* na rzecz *Box Squat* lub ćwiczeń mobilizujących staw skokowy.
- **Realna Dostępność Czasowa:** Pytanie "Ile dni w tygodniu możesz trenować?" jest kluczowe. Jeśli użytkownik deklaruje <3 dni, system musi wymusić plan FBW, gdyż splity przy tak niskiej frekwencji nie zapewniają wystarczającej częstotliwości stymulacji danej partii mięśniowej.<sup>1</sup>

## 4. Anatomia Jednostki Treningowej (Struktura Dnia)

Każdy wygenerowany dzień treningowy musi być spójną logicznie całością, składającą się z trzech nienaruszalnych modułów. Taka struktura zapewnia bezpieczeństwo i optymalizację wydajności.<sup>1</sup>

### 4.1. Moduł 1: Rozgrzewka Specyficzna (Protokół RAMP)

System nie może ograniczać się do ogólnego polecenia "Rozgrzej się". Obiekt JSON musi definiować **Blok Rozgrzewkowy** skorelowany z główną częścią treningu.

- **Raise (Podniesienie):** Podniesienie temperatury ciała (np. 5 min cardio).
- **Activate & Mobilize (Aktywacja i Mobilizacja):** Dobór kontekstowy. Jeśli w Bloku Głównym znajdują się Przysiady, rozgrzewka musi zawierać mobilizację bioder (np. *90/90 Stretch*) i aktywację pośladków (np. *Glute Bridges*).
- **Potentiate (Potencjacja):** Serie rozgrzewkowe (*Warm-up Sets*). Algorytm musi generować serie z narastającym ciężarem, ale niską liczbą powtórzeń (np. 50% x 10, 70% x 5, 90% x 2), aby przygotować układ nerwowy do ciężkich serii roboczych bez wywoływania zmęczenia metabolicznego.<sup>1</sup>

### 4.2. Moduł 2: Blok Główny (Main Workflow)

Serce jednostki treningowej, gdzie realizowane są główne cele adaptacyjne.

- **Struktura:** Może zawierać pojedyncze ćwiczenia (Straight Sets) lub struktury złożone (Superserie, Obwody).
- **Logika Superserii:** Obsługa superserii jest kluczowa dla nowoczesnych aplikacji. Wymaga to zagnieżdżonej struktury danych, gdzie ćwiczenia są grupowane w pary (np. Agonista-Antagonista: Klatka + Plecy), co pozwala na zwiększenie gęstości treningowej.<sup>1</sup>

### 4.3. Moduł 3: Wyciszenie (Cool-down)

- **Cel:** Aktywacja układu przywspółczulnego i obniżenie tętna.
- **Treść:** Rozciąganie statyczne mięśni, które były najbardziej zaangażowane w sesji. Czas trwania pozycji: 30–60 sekund w celu przywrócenia spoczynkowej długości mięśnia.

## 5. Architektura Danych: Schemat JSON

Aby obsłużyć funkcje takie jak "Live Workout Tracking", praca offline i zaawansowane metody treningowe (superserie), model danych musi być rygorystyczny i modularny. Płaskie struktury danych są

odrzućane na rzecz struktur zagnieżdżonych, które wiernie odwzorowują relacje między ćwiczeniami.<sup>7</sup>

Poniżej przedstawiono wzorcowy schemat JSON, będący standardem dla zespołu backendowego i frontendowego (Flutter/React Native).

## 5.1. Globalny Obiekt Planu (WorkoutPlan)

Definiuje metadane całego mezocyklu.

JSON

```
{
  "plan_id": "uuid-v4-plan-001",
  "plan_name": "Hypertrophy Foundations - Phase 1",
  "version": "1.0.0",
  "created_at": "2026-01-10T20:00:00Z",
  "generated_by": "AI_Engine_v2_Claude_Sonnet",
  "user_profile_snapshot": {
    "user_id": "user_123",
    "level": "intermediate",
    "primary_goal": "hypertrophy",
    "injuries": ["lower_back_pain"],
    "available_equipment": ["gym_full"]
  },
  "configuration": {
    "duration_weeks": 4,
    "weekly_frequency": 4,
    "periodization_model": "undulating_non_linear"
  },
  "schedule": // Tablica obiektów WorkoutDay
}
```

## 5.2. Obiekt Dnia Treningowego i Logika Superserii

Kluczowym rozwiązaniem jest zastosowanie `main_workout_blocks`. Blok może zawierać jedno ćwiczenie (zwykła seria) lub listę ćwiczeń (superseria/obwód). Rozwiązuje to problem reprezentacji złożonych struktur treningowych w interfejsie użytkownika.

JSON

```
{
  "day_number": 1,
  "day_type": "training", // Wartości: "training", "rest", "active_recovery"
  "title": "Upper Body Push Focus",
  "description": "Nacisk na klatkę piersiową i tricepsy z elementami siły.",
  "estimated_duration_min": 65,
  "target_muscle_groups": ["chest", "shoulders", "triceps"],

  "warmup": {
    "duration_min": 10,
    "type": "dynamic_contextual",
    "exercises":
  },

  "main_workout_blocks":
    },
    {
      "set_type": "working",
      "reps_min": 6,
      "reps_max": 8,
      "load": { "type": "rir", "value": 2 },
      "tempo": "2010",
      "rest_seconds": 180
    }
  ]
}

],
// BLOK 2: Superseria (Antagonistyczna) - Zwiększenie Gęstości
{
  "block_id": "blk_002",
  "block_type": "superset",
  "order": 2,
  "description": "Superseria antagonistyczna Plecy-Klatka",
  "cycles": 3, // Wykonaj parę 3 razy
  "rest_between_exercises_seconds": 15, // Czas przejścia
  "rest_after_cycle_seconds": 90, // Przerwa po całej parze
  "exercises":
}

],

"cooldown": {
  "duration_min": 5,
  "exercises":
```

```
}  
}
```

### 5.3. Analiza Kluczowych Atrybutów Danych

- **block\_type**: Atrybut krytyczny dla warstwy UI. Wartość "superset" informuje frontend, że ćwiczenia wewnątrz bloku powinny być zgrupowane wizualnie lub wyświetlane w sekwencji naprzemiennej, a nie jedno po drugim do ukończenia wszystkich serii.<sup>9</sup>
- **tempo (np. "3010")**: Explicitna kontrola czasu pod napięciem (Time Under Tension).
  - Cyfra 1: Faza ekscentryczna (opuszczanie).
  - Cyfra 2: Pauza w rozciągnięciu (dół).
  - Cyfra 3: Faza koncentryczna (podnoszenie).
  - Cyfra 4: Pauza w skurczu (góra).
  - *Uzasadnienie*: Dla początkujących, wymuszenie wolnej fazy ekscentrycznej (np. 3-4 sekundy) jest kluczowe dla nauki kontroli motorycznej i prewencji urazów.<sup>1</sup>
- **Obiekt load (Polimorfizm)**: Pozwala na elastyczność preskrypcji obciążenia.
  - type: "rpe" (Skala 1-10) – Najlepsze rozwiązanie dla ogółu populacji.
  - type: "percentage\_1rm" (0.0-1.0) – Dla trójboistów znających swoje maksima.
  - type: "rir" (Reps In Reserve) – Nowoczesny standard, np. "Zatrzymaj się 2 powtórzenia przed upadkiem".<sup>1</sup>
- **reps\_min / reps\_max**: Zamiast sztywnej liczby (np. "10"), stosujemy zakresy. Pozwala to na **Autoregulację**. Jeśli użytkownik z łatwością wykona górną granicę zakresu, jest to sygnał dla algorytmu do progresji ciężaru w kolejnej sesji.<sup>1</sup>

## 6. Algorytmiczna Progresja i Pętla Sprzężenia Zwrotnego (Live Feedback Loop)

Prawdziwa "inteligencja" aplikacji objawia się nie przy generowaniu pierwszego planu, ale w jego adaptacji na podstawie danych z "Live Tracking". Jest to mechanizm **Progresywnego Przeciążenia** (*Progressive Overload*).

### 6.1. Zasada "2-for-2" (Standard NSCA)

Algorytm musi zaimplementować regułę "2-for-2" jako podstawowy mechanizm liniowej progresji.

- **Logika**: Jeśli użytkownik wykona o 2 lub więcej powtórzeń więcej niż założony cel w ostatniej serii danego ćwiczenia podczas dwóch kolejnych treningów, obciążenie dla tego ćwiczenia powinno zostać zwiększone w następnej sesji.
- **Implementacja (Skok Obciążenia)**:
  - Górna Połowa Ciała: Wzrost o 2–5% (ok. 1–2.5 kg).
  - Dolna Połowa Ciała: Wzrost o 5–10% (ok. 2.5–5 kg).<sup>2</sup>

### 6.2. Autoregulacja poprzez RPE/RIR

Dyspozycja dnia (*Daily Readiness*) ulega wahaniom. System używa wskaźnika RPE do dynamicznej korekty objętości w czasie rzeczywistym.



- **Scenariusz:** Użytkownik ma zaplanowane Przysiady 100kg na 3 serie po 5 powtórzeń @ RPE 8.
- **Feedback:** Po pierwszej serii użytkownik raportuje "RPE 10" (Maksymalny wysiłek/Upadek).
- **Akcja Algorytmu:** System wykrywa rozbieżność (Cel RPE 8 vs Rzeczywiste RPE 10). Natychmiast uruchamia protokół **Back-off Set**, redukując obciążenie o 5-10% dla pozostałych serii, aby zapobiec przetrenowaniu i kontuzji. To jest właśnie inteligencja "Live Tracking".<sup>1</sup>

### 6.3. Zarządzanie Stagnacją (Plateaus)

Jeśli użytkownik nie notuje progresu w ciężarze lub powtórzeniach przez 3 kolejne sesje danego ćwiczenia, algorytm powinien podjąć kroki naprawcze:

1. **Analiza Objętości:** Czy całkowita objętość jest zbyt wysoka i uniemożliwia regenerację? (Akcja: Redukcja liczby serii).
2. **Analiza Intensywności:** Czy intensywność jest zbyt wysoka? (Akcja: Wdrożenie "Deload Week" – redukcja obciążeń o 50% na jeden tydzień).
3. **Rotacja Ćwiczeń:** Zamiana ćwiczenia na wariant biomechanicznie zbliżony (np. Zamiana *Flat Bench Press* → *Incline Dumbbell Press*) w celu przełamania adaptacji.<sup>1</sup>

## 7. Protokoły Bezpieczeństwa i Filtrowanie Medyczne (Hard Rules)

System musi działać w ramach sztywnych granic bezpieczeństwa. Są to reguły nadrzędne ("Hard Rules"), których silnik generatywny AI nie może nadpisać ani ignorować.

### 7.1. Przesiew Medyczny (Screening PAR-Q+)

Przed wygenerowaniem jakiegokolwiek JSON-a, użytkownik przechodzi przez filtr oparty na kwestionariuszu *Physical Activity Readiness Questionnaire*.<sup>1</sup>

- **Przeciwwskazania Bezwzględne:** Ból w klatce piersiowej przy wysiłku, niekontrolowane zawroty głowy, niedawny incydent sercowy.
  - **Akcja:** **BLOKADA** generowania planu. Komunikat: "Wymagana konsultacja lekarska przed rozpoczęciem programu."
- **Przeciwwskazania Względne:** Bóle stawowe, kontrolowane nadciśnienie.
  - **Akcja:** **MODYFIKACJA** parametrów planu (Substytucja ćwiczeń, limit RPE).

### 7.2. Logika dla Specjalnych Populacji

Algorytm musi dynamicznie modyfikować bazę dostępnych ćwiczeń w zależności od profilu medycznego:

- **Nadciśnienie (Hypertension):**
  - **Ograniczenie:** Unikanie ćwiczeń z głową poniżej poziomu serca (*Decline Bench Press*, *Decline Sit-ups*).
  - **Ograniczenie:** Unikanie długich izometrii >60s (np. Planki), które gwałtownie podnoszą ciśnienie krwi.
  - **Wskazówka Techniczna:** Wymuszenie w JSON instrukcji "Oddychaj ciągle, unikaj manewru

Valsalvy".<sup>1</sup>

- **Ciąża (Po I trymestrze):**

- *Ograniczenie:* Usunięcie wszystkich ćwiczeń w leżeniu tyłem (supine position), aby zapobiec uciskowi na żyłę główną dolną.
- *Substytucja:* *Bench Press*  $\rightarrow$  *Incline Bench Press* (ławka skośna) lub *Seated Machine Press*.<sup>1</sup>

- **Ból Dolnego Odcinka Pleców (LBP - Lower Back Pain):**

- *Ograniczenie:* Eliminacja wysokiego obciążenia osiowego kręgosłupa (kompresja).
- *Substytucja:* *Barbell Back Squat*  $\rightarrow$  *Split Squat*, *Lunges* lub *Leg Press*.
- *Substytucja:* *Conventional Deadlift*  $\rightarrow$  *Chest-Supported Row* lub *Glute Bridge*.<sup>1</sup>

## 8. Inżynieria Promptów dla AI (LLM Integration)

Aby zmusić Duże Modele Językowe (LLM), takie jak Claude 3.5 Sonnet czy GPT-4, do generowania poprawnego formatu JSON zgodnego z powyższymi regułami, należy stosować zaawansowane techniki inżynierii promptów.

### 8.1. Konstrukcja Promptu Systemowego (System Prompt)

Prompt systemowy musi jednoznacznie definiować "personę" oraz ograniczenia schematu.<sup>1</sup>

Rola: Jesteś ekspertem trenerem przygotowania motorycznego certyfikowanym przez NSCA i ACSM.

Zadanie: Wygeneruj 4-tygodniowy mezocykl treningowy w ścisłym formacie JSON, zgodnie z dostarczonym schematem.

Ograniczenia (Constraints):

1. Zwracaj WYŁĄCZNIE poprawny kod JSON. Żadnego tekstu markdown poza blokiem kodu.
2. Przestrzegaj hierarchii: Najpierw ćwiczenia wielostawowe (Compound), potem izolowane.
3. Wymuszaj czasy przerw: >180s dla siły (<6 powt.), 60-90s dla hipertrofii (6-12 powt.).
4. Używaj wyłącznie `exercise_id` z dostarczonej listy dozwolonych wartości.
5. Jeśli w profilu użytkownika występuje 'lower\_back\_pain', NIE uwzględniaj ćwiczeń: Squat, Deadlift, Bent-over Row.

### 8.2. Techniki "Chain of Thought" i "One-Shot Learning"

Aby zapobiec halucynacjom (np. wymyślaniu nieistniejących ćwiczeń jak "Reverse Gravity Press"), stosujemy:

1. **One-Shot Learning:** Dostarczenie w prompcie pełnego, poprawnego przykładu obiektu JSON reprezentującego jeden dzień treningowy. Model uczy się struktury na podstawie tego wzorca.<sup>14</sup>
2. **Chain of Thought (Łańcuch Myśli):** Wymuszenie na modelu, aby najpierw "przemyślał" (w ukrytym bloku lub pierwszym etapie) ograniczenia użytkownika i wybór podziału treningowego, a dopiero potem wygenerował JSON. Zwiększa to logiczną spójność planu.<sup>16</sup>

### 8.3. Obsługa Superserii w Promptach

Należy explicite poinstruować LLM, jak strukturyzować main\_workout\_blocks dla superserii:

"Grupuj antagonistyczne grupy mięśniowe (np. Klatka i Plecy) w bloku typu 'superset'.  
Ustaw rest\_between\_exercises\_seconds na niską wartość (np. 15s) a  
rest\_after\_cycle\_seconds na wartość regeneracyjną (np. 90s).".<sup>6</sup>

## 9. Podsumowanie i Weryfikacja

Przedstawiona specyfikacja architektoniczna unifikuje rygorystyczne wymagania fizjologii wysiłku z precyzją inżynierii danych. Jest to gotowy plan działania ("blueprint") dla zespołu deweloperskiego.

#### Weryfikacja Kompletności Kryteriów:

1. **Segmentacja Użytkownika:** Zdefiniowana ilościowo (Novice/Int/Adv) wraz z dedykowaną logiką częstotliwości i objętości.<sup>1</sup>
2. **Struktura Treningu:** Wymuszona modułowość (Rozgrzewka -> Blok Główny [Compound->Iso] -> Wyciszenie).<sup>1</sup>
3. **Model Danych JSON:** Zwalidowany, modularny schemat obsługujący superserie, parametry RPE/RIR, Tempo oraz metadane.<sup>17</sup>
4. **Bezpieczeństwo:** Zintegrowane protokoły przesiewowe PAR-Q+ i "twarde reguły" wykluczające dla specyficznych populacji (Cięża, LBP).<sup>1</sup>
5. **Logika Progresji:** Zdefiniowane algorytmy "2-for-2" oraz autoregulacji opartej na RPE dla funkcji Live Tracking.<sup>2</sup>

#### Potwierdzenie Jednoznaczności:

Zaproponowany schemat JSON oraz przepływy logiczne są deterministyczne. Zastosowanie konkretnych typów danych (enum dla typów bloków, zakresy liczb całkowitych dla przerw) eliminuje niejednoznaczność interpretacyjną, gwarantując, że aplikacja będzie zachowywać się jak kompetentny, bezpieczny i "inteligentny" trener. Dokument ten jest gotowy do bezpośredniej translacji na logikę backendową (Python/Node.js) oraz implementację frontendową (Flutter/React Native).