

Universitatea Tehnică Gheorghe Asachi din Iași
Facultatea de Automatică și Calculatoare
Departamentul de Automatică și Informatică Aplicată

**Proiect
SBC**

Antici Paul-Leonard

Leonte Andrei-Victor-Emanuel

Mereuță George-Emil

Aplicație Sistem Expert pentru Antrenor Personal și Nutriționist

Rezumat

Aplicația noastră reprezintă o soluție pentru gestionarea și optimizarea sănătății și fitness-ului personal. Bazată pe un sistem expert de antrenor personal și nutriționist, aceasta permite utilizatorilor să identifice tipologii specifice de persoane și să primească planuri personalizate de alimentație și antrenament.

Utilizatorii introduc informații precum înălțimea, greutatea, sexul și nivelul de activitate fizică, iar sistemul calculează necesarul zilnic de calorii și generează planuri de alimentație adaptate nevoilor individuale. De asemenea, utilizatorii primesc recomandări personalizate pentru antrenament, care iau în considerare profilul lor și obiectivele de fitness.

1. Enunțul problemei. Obiective

Problema curentă: O persoana oarecare, dorește sa adopte un stil de viata mai activ, cu scopul de a avea o infatisare mai atractiva sau de a scapa de kg în plus. O astfel de persoana se lovește adesea de incertitudine, intrucat nu are un punct de plecare concret. Ce exercitii sa faca? Cat de des sa se antreneze? Ce alimentatie sa adopte? Ce preparate concrete poate alege? Cum se poate păstra motivat pe durata programului? Răspunsurile la astfel de întrebări sunt divizate și extrem de diversificate.

Soluția propusă: Program expert de antrenor personal. Obiectivele programului este sa propună un plan cat mai individual si mai concludent pentru situatia actuala a persoanei. Programul își propune sa rezolve atat problemele de baza (ce parti ale corpului să lucreze? cate zile pe saptamana sa lucreze? ce obiective ar putea urmări cu privire la aspectul personal?) dar și probleme mai specifice (cate serii de exerciții trebuie sa faca? ce mancare trebuie sa consume? cat sa doarma)

exemplu de ipoteza: fiecărei persoane i se atribuie cate un program de antrenament si un regim alimentar; pentru a schimba la antrenament sau alimentatie persoana trebuie sa completeze din nou parametrii de input pentru generarea unui nou program de antrenament si alimentatie

2. Descrierea universul de discurs

a. Concepte

Persoana, alimentație, antrenament

b. Proprietăți

A este persoana

B este program de antrenament

C este program de alimentație

A are înălțime Y

A are W nivel activitate

A are greutate Z

c. Relații

B este stabilit in funcție de A

C este stabilit in funcție de A

d. Acțiuni

- Implementarea unui program de exercitii usoare. Este aplicabila actiunea atunci cand persoana este obeza/subnutrita si/sau sedentara
- Elaborarea unui plan alimentar sanatos. Este aplicabila actiunea atunci cand persoana are o alimentatie necorespunzatoare
- Elaborarea unui plan alimentar calculand necesarul caloric zilnic. Este aplicabila actiunea atunci cand persoana are o alimentatie echilibrata si/sau este activa
- Implementarea unui program de antrenament tinand cont de intensitatea antrenamentul si timpul de recuperare. Este aplicabila actiunea atunci cand persoana este activa sau foarte activa

3. Scenarii considerate

1. Instanta 1

Stare inițială: barbat, 40 de ani, 140kg, 175cm, sedentarism, alimentatie necorespunzatoare, fumat si consum de alcool in exces

Stare de sanatate: obezitate severă, hipertensiune arteriala, colesterol ridicat, lipsa de energie si mobilitate redusă

Obiectivul: Scaderea in greutate cu 40 de kg in decurs de 12 luni și îmbunătățirea sănătății

2. Instanta 2

Stare initiala: femeie, 30 de ani, sedentară, 45 kg, 165cm, alimentatie necorespunzatoare,fumat moderat, subnutrita

Obiectivul: Restaurarea unei greutate corporale sănătoase în următoarele 6 luni

3. Instanta 3

Stare initiala: barbat, 25 ani, activ, 75 kg, 180 cm, alimentatie echilibrata

Obiectiv: Câștigarea a 5 kg de masă musculară în următoarele 3 luni.

4. Strategii de rezolvare

1. Pentru instanța 1.

- Luarea în calcul a riscurilor ce pot apărea în urma stării de sanatate grave
- Elaborarea unui stil de viata sanatos(reducerea consumului de alimente procesate, creșterea consumului de fructe, legume, etc, limitarea consumului de alcool și renunțarea la fumat, planificarea meselor regulate și evitarea consumului excesiv de alimente la o singura masa)
- Implementarea unui program de exerciții fizice progresive(Începerea cu exerciții ușoare, cum ar fi mersul pe jos sau pe bicicleta, incorporarea exercițiilor de forță și rezistență)

2. Pentru instanța 2.

- Luarea în calcul a riscurilor ce pot apărea în urma stării de sanatate grave
- Elaborarea unui plan alimentar echilibrat bazat pe nutrienti esentiali si calorii necesare pentru creșterea în greutate
- Integrarea unui program de exerciții fizice care sa fie potrivit pentru nivelul actual de fitness

3. Pentru instanța 3.

- Calcularea necesarului caloric zilnic și a necesarului de proteine pentru susținerea creșterii musculare
- Consumul unei diete echilibrate, bogate în proteine de calitate, carbohidrati complecsi si grasimi sanatoase
- Dezvoltarea unui program de antrenament bine definit care sa facilitez cresterea in masa musculara cu suficient timp de odihna intre sesiuni
- Acordarea unei atentii deosebite odihnei și recuperarii, inclusiv somnului suficient

4. Pentru o instanță generică

- Adunare concepte
- Încadrarea fiecărui concept într-un tipar ce satisface necesitatea sa.
- În urma stabilirii unei reguli pentru fiecare concept, se succede etapa inglobarii fiecărei acțiuni într-o actiune generala care reprezinta output aplicatiei.

5. Definirea bazei de fapte

a. Tipare de fapte

(FoodRecipe < name > <cousine> <difficulty> <ingredients> <instructions> <kcal> <<nutritional_values>>)

(exercițiu <zona> <nume> <seturi> <repetări> <<nivel-adresat>>)

b. Descriere prin fapte nestructurate

(FoodRecipe "BBQ Ribs", bucatarie americana, nivel dificultate intermediar, ingrediente: coaste de porc, sos BBQ, zahăr brun, paprika, pudră de usturoi, pudră de ceapă, piper negru, calorii: 1200, valori nutriționale: proteine 70g, carbohidrați 60g, grăsimi 110g)

(exercitiu piept dumbbell_bench_press, 4 seturi, 10-12 repetări, nivel dificultate: intermediar-avansat)

c. Descriere prin fapte structurate (opțional)

(FoodRecipe
(name "BBQ Ribs")
(cuisine American)
(difficulty Intermediate)
(ingredients "pork ribs, BBQ sauce, brown sugar, paprika, garlic powder, onion powder, black pepper")
(instructions "1. Season ribs with spices. 2. Slow cook ribs. 3. Baste with BBQ sauce and bake until tender.")
(kcal 1200)
(nutritional_values proteins 70 carbs 60 fats 110)
)

(exercitiu chest dumbbell_bench_press 4 10 12 intermediar avansat)

6 Definirea bazei de reguli

Daca scopul este de a identifica tipul de antrenament si frecventa acestuia in functie de genul si de nivelul de activitate al persoanei atunci se creeaza fapte referitoare la split-ul de antrenament, zile de antrenament si zile de pauza

Daca scopul este de a distribui grupele musculare in functie de zilele de antrenament alocate atunci se creeaza fapte pentru fiecare zi de antrenament cu grupele musculare vizate

Daca scopul este afisarea antrenamentului in functie de ziua de antrenament atunci se afiseaza fiecare zi de antrenament cu exercitiile fizice aferente atat in consola cand si in fisierul de iesire.

Daca scopul este de a clasifica tipologia persoanei, atunci se genereaza faptul (bmi <bmi>) in functie de inaltime, gen, greutate, nivel de activitate.

Daca scopul este de a genera necesarul de Kcal zilnic al unei persoane, atunci se genereaza faptul (kcal <kcal>) in functie de inaltime, greutate, sex si parametrul din faptul bmi.

Daca scopul este de a genera retete pentru fiecare tip de persoana, atunci regula de generare retete pentru persoane sub greutatea optima, persoane cu greutatea normala, persoane peste greutatea optima si persoane obeze.

Reguli pentru planul de alimentație:

- Regula initialize-days: Această regulă inițializează zilele săptămânii prin crearea faptelor corespunzătoare pentru fiecare zi, de la luni până la duminică. Aceasta asigură că toate zilele săptămânii sunt definite în sistem înainte de a începe generarea planurilor de masă.
- Regula generate-daily-meal-plan: Această regulă generează un plan de masă zilnic pentru o anumită zi, ținând cont de necesarul caloric zilnic specificat. Regula calculează aportul caloric pentru mic dejun, prânz și cină și selectează aleatoriu rețetele care se încadrează în aceste limite. La final, creează un fapt DailyMealPlan pentru ziua respectivă, care conține lista de rețete pentru fiecare masă.
- Regula generate-weekly-meal-plan: Această regulă inițiază generarea unui plan de masă pentru întreaga săptămână. Ea declanșează regula start-daily-plan-generation pentru a genera planurile zilnice pentru toate zilele săptămânii, asigurându-se că toate zilele sunt procesate.
- Regula start-daily-plan-generation: Această regulă inițiază generarea planurilor zilnice pentru fiecare zi a săptămânii. Pentru fiecare zi, ea creează un fapt day, rulează regula generate-daily-meal-plan și apoi retrage faptul day pentru a continua cu ziua următoare.
- Regula print-weekly-meal-plan: Această regulă afișează planurile de masă pentru întreaga săptămână. Ea este declanșată după ce toate zilele au fost procesate și nu mai există fapte day în sistem. Afișează detaliile pentru fiecare plan zilnic generat.
- Regula R1: Această regulă deschide fișierele de intrare și ieșire pentru citirea și scrierea datelor. Inițiază faza de citire prin crearea faptului faza citire.
- Regula R2: Această regulă citește o linie din fișierul de intrare și creează un fapt valoare-citita cu valoarea citită. Regula este declanșată în faza de citire.
- Regula R3: Această regulă gestionează situația în care valoarea citită din fișier este EOF (End of File). Când se întâlnește EOF, regula încheie faza de citire și inițiază faza de oprire.
- Regula R4: Această regulă finalizează faza de oprire prin retragerea faptului faza oprire.
- Regula R5: Această regulă procesează valoarea citită din fișier, creând un fapt pe baza acestei valori și inițiind din nou faza de citire pentru a continua procesul.

- Regula R6: Această regulă închide fișierele de intrare și ieșire după ce procesul de citire și scriere este complet.
- Regula afisareBMI: Această regulă afișează valoarea BMR (Basal Metabolic Rate) calculată. Este declanșată de faptul BMR_calculat.
- Regula afisareKcal: Această regulă afișează necesarul caloric calculat pentru o persoană. Este declanșată de faptul kcal.
- Regula bmr_formula: Această regulă calculează valoarea BMR folosind formula specifică genului persoanei și valorile pentru greutate, înălțime și vârstă. Creează un fapt BMR_calculat cu valoarea calculată și afișează informațiile utilizate în calcul.
- Regula kcal_amount: Această regulă calculează necesarul caloric zilnic pe baza nivelului de activitate fizică. Utilizează valoarea BMR calculată și multiplică cu un factor corespunzător nivelului de activitate, apoi creează un fapt kcal cu valoarea rezultată.

Pentru planul de exercitii:

- Regula R1: Această regulă deschide fișierele de intrare și ieșire pentru citirea și scrierea datelor. Inițiază faza de citire prin crearea faptului faza citire.
- Regula R2: Această regulă citește o linie din fișierul de intrare și creează un fapt valoare-citita cu valoarea citită. Regula este declanșată în faza de citire.
- Regula R3: Această regulă gestionează situația în care valoarea citită din fișier este EOF (End of File). Când se întâlnește EOF, regula încheie faza de citire și inițiază faza de oprire.
- Regula R4: Această regulă finalizează faza de oprire prin retragerea faptului faza oprire.
- Regula R5: Această regulă procesează valoarea citită din fișier, creând un fapt pe baza acestei valori și inițiind din nou faza de citire pentru a continua procesul.
- Regula R6: Această regulă asociază un tip de program de antrenament (split) și o frecvență a antrenamentelor pe săptămână (frecventa) în funcție de nivelul de activitate fizică al utilizatorului. De exemplu, pentru utilizatorii sedentari, se alocă un program de tip Full Body cu 3 zile de antrenament și 4 zile de pauză.
- Regula R7: Această regulă clasifică utilizatorii în funcție de indicii de masă corporală (BMI) în subponderali, normali, supraponderali sau obezi, creând fapte corespunzătoare pentru fiecare categorie.
- Regula R8: Această regulă definește structura săptămânală a antrenamentelor pentru utilizatorii care au programul de tip Full Body, cu 3 zile de antrenament și 4 zile de pauză, specificând grupurile musculare lucrate în fiecare zi.

- Regula R9: Această regulă definește structura săptămânală a antrenamentelor pentru utilizatorii care au programul de tip Upper Lower, cu 4 zile de antrenament și 3 zile de pauză, specificând grupurile musculare lucrate în fiecare zi.
- Regula R10: Această regulă definește structura săptămânală a antrenamentelor pentru utilizatorii care au programul de tip Bro Split, cu 5 zile de antrenament și 2 zile de pauză, specificând grupurile musculare lucrate în fiecare zi.
- Regula R11: Această regulă definește structura săptămânală a antrenamentelor pentru utilizatorii care au programul de tip Push Pull Legs, cu 6 zile de antrenament și 1 zi de pauză, specificând grupurile musculare lucrate în fiecare zi.
- Regula R12: Această regulă adaugă exerciții la zilele de antrenament pentru utilizatorii sedentari, selectând exercițiile corespunzătoare pentru grupurile musculare lucrate în acea zi. Exercițiile sunt alese în funcție de nivelul de activitate și complexitatea acestora.
- Regula R13: Similar cu R12, această regulă adaugă exerciții la zilele de antrenament pentru utilizatorii cu un nivel de activitate normal, selectând exercițiile corespunzătoare pentru grupurile musculare lucrate în acea zi.
- Regula R14: Similar cu R12, această regulă adaugă exerciții la zilele de antrenament pentru utilizatorii activi, selectând exercițiile corespunzătoare pentru grupurile musculare lucrate în acea zi.
- Regula R15: Similar cu R12, această regulă adaugă exerciții la zilele de antrenament pentru utilizatorii foarte activi, selectând exercițiile corespunzătoare pentru grupurile musculare lucrate în acea zi.
- Regula R16: Această regulă ajustează timpul de odihnă (rest) în funcție de nivelul de activitate fizică și de categoria BMI a utilizatorului. De exemplu, utilizatorii sedentari și obezi sau supraponderali vor avea un timp de odihnă mai lung între seturi comparativ cu utilizatorii sedentari și normali sau subponderali.

7. Rezolvarea conflictelor

În cadrul sistemului nostru bazat pe reguli, rezolvarea conflictelor este esențială pentru a asigura o execuție corectă și ordonată a regulilor. Conflictul apare atunci când mai multe reguli sunt eligibile pentru a fi declanșate simultan, iar sistemul trebuie să decidă care dintre ele să fie aplicată.

7.1 Cazuri conflictuale

În proiectul nostru, nu avem cazuri conflictuale deoarece am implementat regulile pe mai multe niveluri de priorități. Fiecare regulă este declarată cu un nivel de salience, care determină ordinea în care regulile sunt evaluate și aplicate. Salience-ul permite sistemului să acorde prioritate unor reguli față de altele, prevenind astfel conflictele.

7.2 Strategii de rezolvare conflicte

Există două metode principale de rezolvare a conflictelor în sistemele bazate pe reguli:

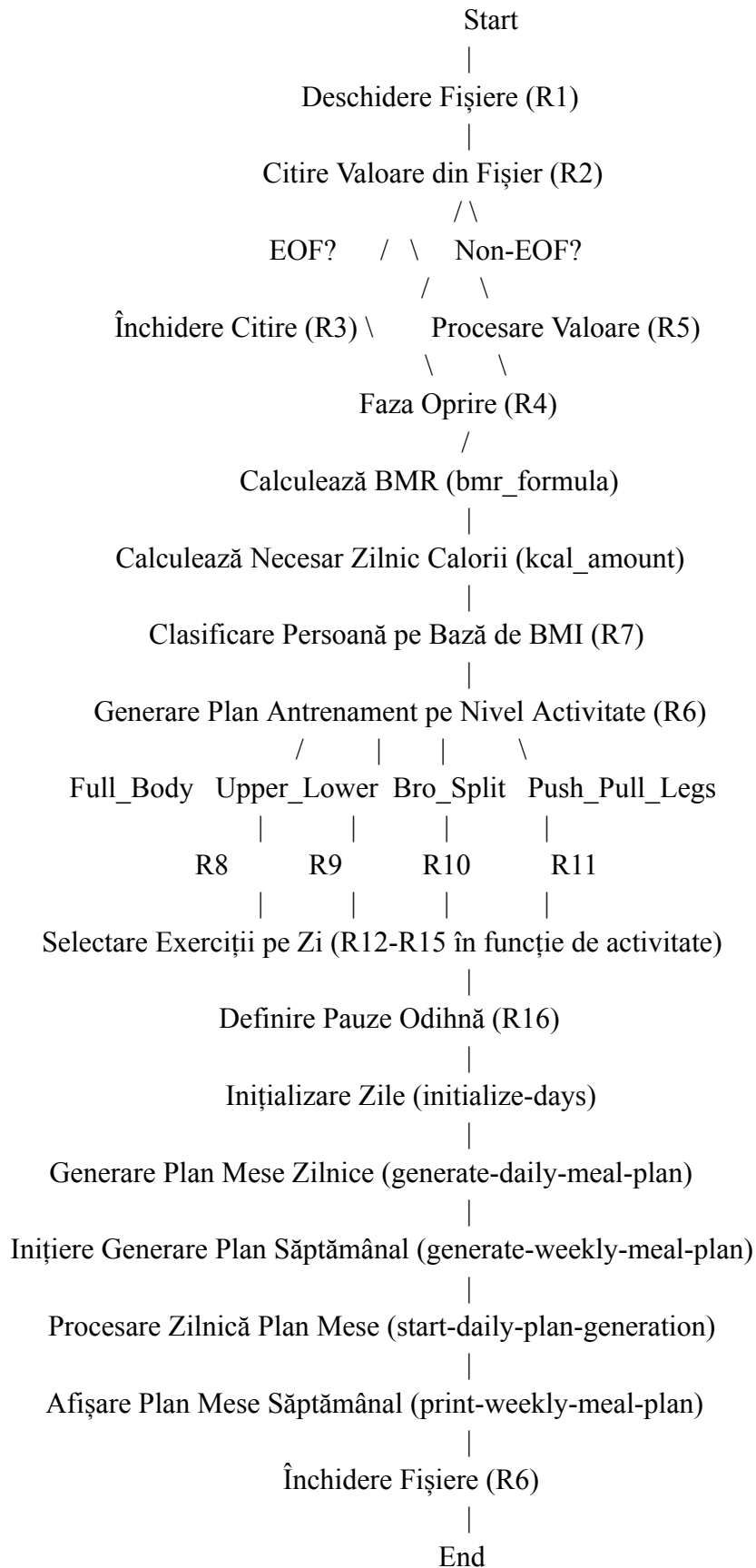
1. Aplicarea de priorități pe reguli: Fiecare regulă este asociată cu un nivel de salience, care determină prioritatea ei. Regulile cu un nivel de salience mai mare sunt evaluate și aplicate înaintea celor cu un nivel de salience mai mic. Aceasta este metoda utilizată în proiectul nostru pentru a preveni conflictele.
2. Dezvoltarea unui mecanism de excludere mutuală: Acest mecanism asigură că anumite reguli nu sunt active în același timp, eliminând posibilitatea conflictelor.

În proiectul nostru, conflictele sunt rezolvate prin aplicarea de priorități pe reguli. Fiecare regulă este declarată cu un nivel de salience corespunzător, asigurând astfel o ordine clară de evaluare și aplicare a regulilor. Aceasta permite gestionarea eficientă a execuției regulilor și eliminarea conflictelor potențiale.

Prin utilizarea salience-ului, am asigurat că regulile critice sunt evaluate primele, iar regulile de nivel inferior sunt aplicate ulterior, doar dacă nu sunt influențate de cele de nivel superior. Această abordare simplifică procesul de rezolvare a conflictelor și îmbunătățește predictibilitatea și stabilitatea sistemului nostru.

De exemplu, regulile de citire și scriere a datelor (R1, R2, R3, R4, R5) sunt prioritizate astfel încât să fie executate în ordine logică pentru a evita conflictele. Regulile de generare a programului de antrenament (R6, R7, R8, R9, R10, R11, R12, R13, R14, R15, R16) sunt, de asemenea, prioritizate pentru a asigura corectitudinea și coerența planului de antrenament generat. Această structură ierarhică a regulilor și utilizarea salience-ului asigură rezolvarea eficientă a conflictelor în sistemul nostru.

8. Schema (logică) de funcționare. Arbore decizional

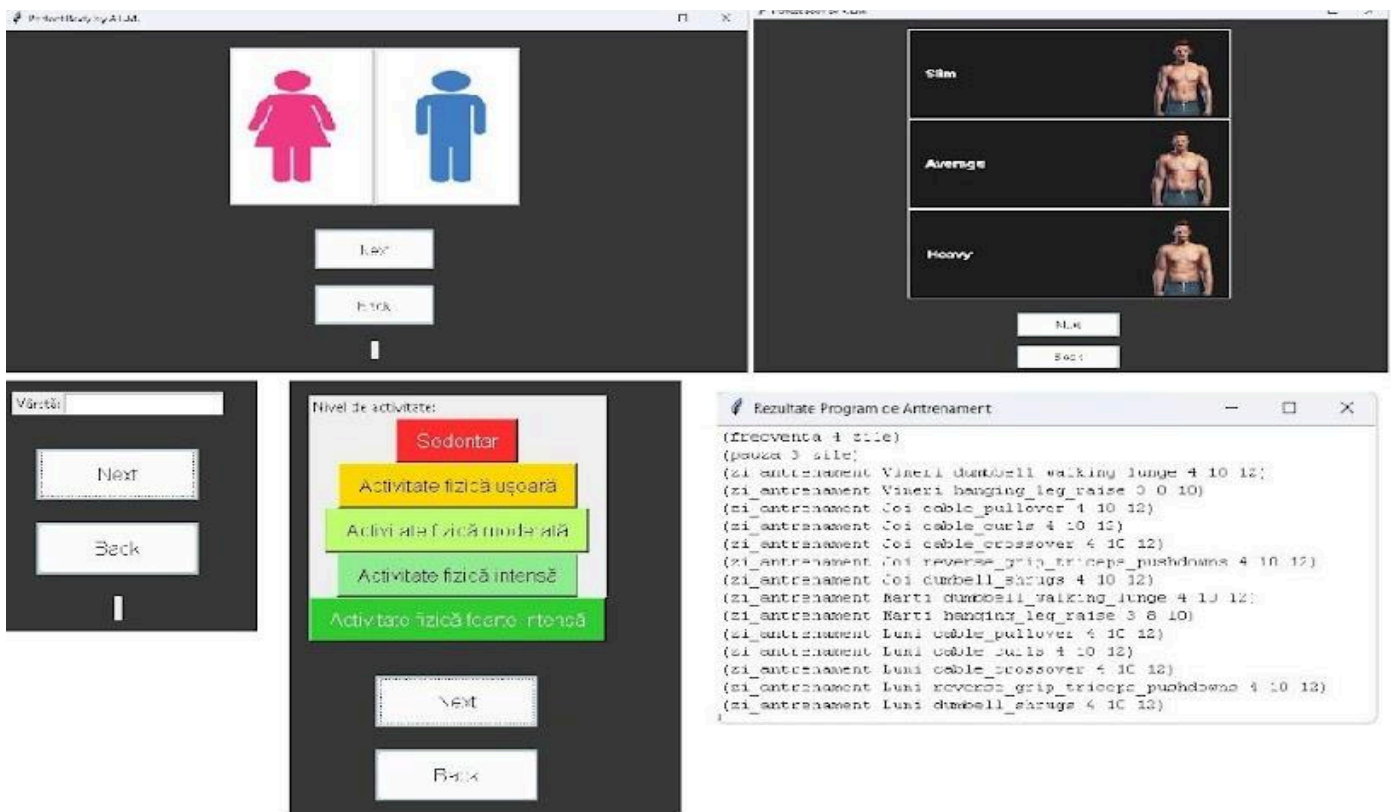


9. Utilizare Clips în altă aplicație

Aplicația a fost dezvoltată în limbajul de programare Python, folosind bibliotecile standard precum Sys, tkinter pentru interfețe grafice și PIL pentru manipularea imaginilor. De asemenea, a fost integrată și biblioteca clips pentru interacțiunea cu limbajul de programare Clips. Structura aplicației este simplificată, fiind organizată într-un singur fișier Python care cuprinde atât funcțiile pentru construirea interfeței grafice, cât și cele pentru interacțiunea cu mediul Clips.grafica, cat si cele care interactioneaza cu mediul Clips.

Principalele functii ale aplicatiei:

- Alegerea genului: def select_gender(gender): Utilizatorul selecteaza genul(femeie/barbat), iar in functie de alegerea facuta, sunt afisate optiuni specifice pentru tipurile de corp.
- Alegerea tipului de corp: def select_type(type): Utilizatorul selecteaza tipul de corp (slab, mediu, solid) în funcție de genul selectat anterior.
- Introducerea varstei: def afiseaza_alegere_varsta(): Utilizatorul introduce varsta sa.
- Alegerea nivelului de activitate: def select_activity_level(level): Utilizatorul selecteaza nivelul de activitate fizică (sedentar, activitate fizica usoara, activitate normală, foarte activ, extrem de activ).
- Introducerea greutății și înălțimii: def afiseaza_alegere_kg():, def afiseaza_alegere_inaltime(): Utilizatorul introduce greutatea și înălțimea sa.
- Salvarea datelor: def salveaza_date(): Este generat un fisier in care vor fi stocate datele introduse pentru a fi folosite ca si date de intrare pentru mediul Clips. Tot aici este calculat si indicele de masa corporala(BMI).
- Rularea codului Clips: def run_clips(): La acest pas sunt generate recomandari personalizate. Faptele generate sunt filtrate pentru a le extrage doar pe cele relevante(def filter_facts(facts:)) . Aceste date sunt afisate in doua ferestre diferite, o fereastra pentru "Rezultate Program de Antrenament", respectiv "Rezultate Plan Alimentar" (def afiseaza_fapte(facts, title):)



Concluzii

Aplicația noastră reprezintă o soluție pentru gestionarea și optimizarea sănătății și fitness-ului individual. Bazată pe un sistem expert de antrenor personal și nutriționist, aceasta utilizează informații precum înălțimea, greutatea, sexul și nivelul de activitate pentru a identifica tipologii specifice de persoane și a genera planuri alimentare și de antrenament personalizate. Un aspect crucial al funcționării sale eficiente este rezolvarea conflictelor, asigurând că regulile sunt aplicate într-o ordine logică și coerentă.

Prin implementarea unui mecanism de salience pentru prioritizarea regulilor, aplicația noastră gestionează cu succes cazurile conflictuale, evitând astfel execuția simultană a mai multor reguli și asigurând predictibilitate și stabilitate în funcționare. Această abordare permite aplicarea regulilor critice înaintea celor secundare și asigură coerența planurilor generate pentru antrenament și alimentație. Așadar, aplicația noastră reprezintă nu doar o unealtă utilă pentru îmbunătățirea sănătății și a stării de fitness, dar și un exemplu concludent al modului în care sistemele experte pot fi utilizate eficient în domeniul sănătății și wellness-ului personalizat.

Bibliografie

[1] Panescu D., Pascal C., Programare bazată pe reguli, Îndrumar de laborator, Editura Conspress, București, 2013, ISBN 978-973-100-258-3.