UNIVERSITATEA „ALEXANDRU-IOAN CUZA” DIN IAȘI

**FACULTATEA DE INFORMATICĂ**

|  |
| --- |
|  |

LUCRARE DE LICENȚĂ

**VivaFit**

propusă de

**Bejan Paul – Eusebiu**

**Sesiunea:** iulie, 2025

Coordonator științific

**Drd. Ioniță Alexandru**

UNIVERSITATEA „ALEXANDRU-IOAN CUZA” DIN IAȘI

**FACULTATEA DE INFORMATICĂ**

LUCRARE DE LICENȚĂ

**VivaFit**

propusă de

**Paul – Eusebiu Bejan**

**Sesiunea:** iulie, 2025

Coordonator științific

**Drd. Ioniță Alexandru**

**Anexa IV**

Avizat,

Îndrumător lucrare de licență,

Drd. Ioniță Alexandru.

Data: ............................ Semnătura: ............................

**Declarație privind originalitatea conținutului lucrării de licență**

Subsemnatul **Bejan Paul - Eusebiu** domiciliat în **România, jud. Iași, mun. Iași, strada Vasile A. Urechia, nr. 4, bl. M6, et. 1, ap. 11**, născut la data de **29 noiembrie 2003**, identificat prin CNP **5031129226703** , absolvent al **Universității „Alexandru Ioan - Cuza” din Iași, Facultatea de Informatică** specializarea **Informatică**, promoția **iulie 2025**, declar pe propria răspundere, cunoscând consecințele falsului în declarații în sensul art. 326 din Noul Cod Penal și dispozițiile Legii Educației Naționale nr. 1/2011 art.143 al. 4 si 5 referitoare la plagiat, că lucrarea de licență cu titlul **Vivafit** elaborată sub îndrumarea domnului **Drd. Ioniță Alexandru**, pe care urmează să o susțină în fața comisiei este originală, îmi aparține și îmi asum conținutul său în întregime.

De asemenea, declar că sunt de acord ca lucrarea mea de licență să fie verificată prin orice modalitate legală pentru confirmarea originalității, consimțind inclusiv la introducerea conținutului său într-o bază de date în acest scop.

Am luat la cunoștință despre faptul că este interzisă comercializarea de lucrări științifice in vederea facilitării falsificării de către cumpărător a calității de autor al unei lucrări de licență, de diploma sau de disertație și în acest sens, declar pe proprie răspundere că lucrarea de față nu a fost copiată ci reprezintă rodul cercetării pe care am întreprins-o.

Data: ………………………… Semnătură student: …………………………

**Anexa V**

**Declarație de consimțământ**

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul **VivaFit**, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” din Iași, să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Absolvent **Paul – Eusebiu Bejan**

Data: ………………………… Semnătură student: …………………………

**Abstract**

Această lucrare prezintă atât funcționalitățile, cât și implementarea, dezvoltarea și arhitectura aplicației **VivaFit**, o **platformă web** a cărei principalul scop este gestionarea stilului de viață sănătos prin urmărirea alimentelor consumate și a exercițiilor fizice realizate. Aplicația oferă utilizatorilor posibilitatea de a își organiza mesele zilnice, să creeze și să vizualizeze alimente sau rețete, să își noteze exercițiile fizice realizate pe parcursul zilei, să își noteze consumul de apă și masa corporală zilnic pentru a își atinge obiectivele dorite.

Un element important al programului îl constituie componenta de **BMR (Basal Metabolic Rate)** care pune la dispoziție utilizatorilor posibilitatea de a își calcula în funcție de anumite caracteristici caloriile și macronutrienții necesari, având posibilitatea de a ajusta manual ulterior valorile pentru a își seta propriile obiective.

De asemenea, aplicația oferă utilizatorilor posibilitatea de a își realiza cont pe platformă în funcție de rolul dorit (**Utilizator obișnuit**, **Nutriționist** și **Antrenor**). Atât antrenorii, cât și nutriționiștii au la dispoziție funcționalitatea de a își crea o pagină personalizată în cadrul aplicației, unde își pot promova profesia și serviciile.

O altă caracteristică a platformei este integrarea unui sistem de conversație cu inteligența artificială prin intermediul uneltei **Ollama**, care este capabil să propună informații personalizate în sfera de conversație a nutriției, fitness-ului și alimentației. În plus, acest sistem are funcționalitatea de a analiza o masă introdusă de utilizator in aplicație, oferind sfaturi legate de alimentele și macronutrienții consumați în funcție de tipul mesei și obiectivele utilizatorului.

Un alt aspect inovativ al aplicației îl constituie funcționalitatea de estimare a numărului de calorii arse în timpul unui antrenament, pe baza unor anumiți indici (precum datele personale și nivelul de intensitate). Această caracteristică este posibilă prin implementarea unei **rețele neuronale** care reușește, pe baza unei mulțimi de date, să estimeze caloriile arse cu o rată mică de eroare, oferind utilizatorilor facilitatea de a își urmări mai în detaliu activitatea fizică.

Pentru îmbunătățirea continuă a aplicației, dar și a datelor despre alimente și rețete, platforma beneficiază de un formular de contact, prin care orice persoană poate să intre în legătura cu echipa de administratori. Această funcționalitate facilitează comunicarea deschisă și permanentă cu utilizatorii aplicației, prin care se dorește crearea unei comunități active implicată în evoluția și optimizarea platformei, contribuind astfel la un nivel ridicat de mulțumire a utilizatorilor prin adaptarea la cerințele reale ale comunității.

Din punct de vedere al tehnologiilor folosite, aplicația implementează partea de interfață a utilizatorului prin Angular, iar partea de logică și procesare a datelor se realizează prin **Spring Boot** și **Python**, implementând o arhitectură de tip **RESTful API**, care permite performanță și scalabilitate. Totodată, partea de stocare și gestionare a mulțimilor de date se implementează prin folosirea unei baze de date relațională **MariaDB**, a cărui scop este salvarea informațiilor despre utilizatori, mesele, alimentele, rețetele, datele nutriționale asociate și alte aspecte.

Așadar, având în vedere funcționalitățile diverse implementate, aplicația **VivaFit** pune la dispoziție un instrument integral pentru realizarea unui stil de viață sănătos prin gestionarea meselor și a exercițiilor personalizate, dar și prin analiza inteligentă a consumului de alimente realizat și a sfaturilor oferite.

*Cuvinte cheie: VivaFit, Angular, Spring Boot, Python, RESTful API, MariaDB, Ollama, BMR, aplicație Web, rețea neuronală, nutriție, fitness*

**Cuprins**

# Introducere

## Motivație

Aplicația VivaFit a fost concepută din cauza unei probleme profunde personale. Încă de când eram mic, fiind o persoană pofticioasă, cu obiceiuri alimentare haotice și tentat să mănânc mult și nesănătos, am avut dificultăți mari în a avea o creștere sănătoasă și lipsită de kilograme în plus. De cele mai multe ori, în încercarea de a consuma o cantitate mai mică de mâncare, pentru a îmi mențiune masa corporală sau a slăbi, înlocuiam anumite alimente considerate de mine dens calorice cu altele mai slab calorice. Problema era că niciodată nu îmi măsuram cantitatea alimentelor consumate, astfel deși eu îmi substituiam alimentele, din nevoia de a îmi satisface foamea, ajungeam să consum o cantitate mai mare din alimentul slab caloric, iar în final avem ingerate același număr sau mai multe calorii. Așadar, eu, în majoritatea zilelor, nu reușeam să îmi ating obiectivul datorită faptului că nu aveam cunoștințe despre câte calorii trebuie să consum, cum să îmi împart aportul energetic în macronutrienți (proteine, carbohidrați, grăsimi) și să îmi estimez numărul de calorii arse prin intermediul activităților fizice realizate.

Având în vedere dificultățile personale și faptul că multe persoane se combat cu ele, trebuie evidențiat că „*Fiind a doua cauză de deces care poate fi prevenită (după fumat), obezitatea reprezintă o problemă majoră de sănătate publică. În prezent, mai mult de 1 miliard de oameni au surplus de masă corporală (supraponderalitate) și peste 300 milioane suferă de obezitate. Mai mult decât atât, în următoarele două decenii numărul se poate dubla, fapt ce va duce la o creștere semnificativă a patologiei asociate, iar durata medie de viață a pacienților cu obezitate este cu 8-10 ani mai scurtă decât a persoanelor cu masă corporală normală.*” (1)

Observând aspectele negative introduse în sănătatea mea generală, provocate de stilul meu de viață imprevizibil și de riscurile alarmante declanșate de obezitate, a fost necesar să găsesc o soluție eficace și abordabilă în vederea dorinței de a îmi prelua controlul asupra propriei vieți și a obiceiurilor zilnice. Inexistența unei unelte eficiente de monitorizare a alimentației, dar și a exercițiilor fizice, m-au oprit din a îmi atinge scopurile legate de calitatea vieții. Din această cauză, am optat să implementez o aplicație plurivalentă[[1]](#footnote-1), care să îmi permită să am o imagine de ansamblu asupra progresului meu, astfel încât să pot să realizez ajustări informate și adecvate, stimulând astfel o prelucrare trainică a stilului de viață echilibrat.

## Soluția propusă

Soluția gândită și recomandată de mine este aplicația web VivaFit, care este ușor de utilizat, accesibilă atât pe calculatoare și laptopuri, cât și pe alte dispozitive precum telefoane și tablete, prin intermediul căreia utilizatorii își pot gestiona și urmări zilnic mesele, consumul de apă și exercițiile fizice. Avantajele implementării unei aplicații web și nu a unui program nativ constau în rapiditatea de accesare, economisirea spațiului de stocare a dispozitivului și utilizarea acesteia indiferent de platforma pe care rulează.

Aplicația oferă posibilitatea de a calcula necesarul caloric zilnic în funcție de caracteristicile personale (gen, masă corporală, vârstă, înălțime, nivel de activitate, obiectiv), permite urmărirea caloriilor și a macronutrienților atât la nivel zilnic, cât și la nivelul meselor, dar și a propriei greutăți. Totodată, utilizatorii își pot crea propriile alimente sau rețete cu caloriile și macronutrienții definiți de ei, pe care ulterior le împărtășesc cu restul comunității.

De asemenea, implementarea secțiunii de conversație cu inteligența artificială vine în sprijinul utilizatorilor pentru oferirea de sfaturi rapide în legătură cu nutriția și activitatea fizică, iar funcția de analiză a alimentelor și a macronutrienților dintr-o masă poate îmbunătăți deciziile viitoare ale utilizatorilor.

În plus, pentru a oferi susținere celor care, asemenea mie, nu cunosc în permanență sau nu dețin dispozitive care estimează numărul de calorii arse, aplicația implementează o funcționalitate de apreciere a numărului de calorii pierdute în cadrul unei activități fizice, prin intermediul anumitor informații (caracteristici personale, nivelul de antrenament, durata), furnizând utilizatorilor o imagine de ansamblu clară prin care aceștia își pot ajusta informat detaliile necesare atingerii obiectivului propus.

Mai mult decât atât, utilizatorii au oportunitatea de a genera rapoarte detaliate și grafice interactive, prin care își pot monitoriza aportul de macronutrienți pe parcursul mai multor zile, precum și greutatea corporală, pentru a susține menținerea motivației pe termen lung și pentru a urmări evoluția față de așteptările setate.

## Impact și beneficii

O aplicație precum VivaFit are o sferă de întrebuințări destul de ridicată în viața de zi cu zi, soluționând mai multe aspecte și dificultăți cu care se înfruntă oamenii:

* Înlesnește accesul rapid de pe orice dispozitiv, fără a fi necesară instalarea altor componente, prin intermediul platformei web, ducând la crearea unei experiențe flexibile și convenabile.
* Inițiază însușirea unui stil de viață sănătos, punând la dispoziție funcționalitățile necesare precum urmărirea alimentației, hidratării, activităților fizice și masei corporale.
* Micșorează periculozitățile asociate sedentarismului, dar și a obezității, prin stimularea unei alimentații echilibrate și a antrenamentelor fizice regulate.
* Oferă opțiunea calculului personalizat al necesarului caloric zilnic și al macronutrienților în vederea atingerii obiectivelor personale, ținând cont de măsurătorile fizice ale persoanei.
* Pune la dispoziție vizualizarea graficelor și rapoartelor, prin intermediul cărora utilizatorii își pot monitoriza progresul atât pe termen scurt, cât și pe termen lung, reglându-și tactica în funcție de rezultate.
* Furnizează posibilitatea de a scana codul de bare al alimentelor sau ingredientelor, cu camera dispozitivului, pentru a îl introduce mai ușor în cadrul jurnalului alimentar sau pentru a asocia codul unui produs.
* Implementează o secțiune de inteligență artificială prin care se pot oferi recomandări nutriționale pe baza analizei unei mese și poate estima numărul de calorii pierdute în timpul unui antrenament.
* Facilitează dobândirea treptată a cunoștințelor și principiilor unui stil de viață echilibrat, prin intermediul informațiilor clare despre densitatea calorică a alimentelor.

# Aplicații similare

La nivelul pieței aplicațiilor pentru monitorizarea consumului alimentar și a modului de viață sănătos există deja diferite platforme cunoscute precum „MyFitnessPal”, „Eat & Track” și „Cronometer”. Aceste aplicații pun la dispoziție utilizatorilor posibilitatea de a gestiona aportul caloric, distribuția macronutrienților, precum și greutatea corporală, având ca fundament folosirea de mulțimi de date extinse cu alimente și rețete. Cu toate acestea, aceste soluții implementate prezintă anumite limitări în ceea ce privește interacțiunea și integrarea cu tehnologii contemporane care să sprijine experiența utilizatorilor în timp real.

Aplicația VivaFit, propusă în această lucrare, furnizează un plus substanțial prin implementarea unor funcționalități sofisticate care nu se regăsesc în soluțiile menționate mai sus. Prin urmare, VivaFit include o componentă de conversație cu inteligența artificială (Ollama AI) care are ca scop atât analiza și recomandarea unor alternative de alimente, în funcție de obiectivul fiecărui utilizator, în cazul meselor introduse, cât și a opțiunii de a avea conversații libere legate de nutriție și exerciții fizice. Mai mult decât atât, prin intermediul unei rețele neuronale, aplicația pune la dispoziție utilizatorilor funcționalitatea de estimare a caloriilor arse pe parcursul unei activități fizice.

Un alt beneficiu considerabil al aplicației propuse este integrarea cu cântare inteligente Bluetooth (modele OkOk International), care oferă utilizatorilor funcționalitatea de a își nota greutatea corporală direct în aplicație, simplificând procedura de supervizare a datelor personale. În plus, există opțiunea de a vizualiza predicții personalizate asupra evoluției greutății corporale în intervalul următor de zile, pe baza înregistrărilor din trecut privind masa corporală. Totodată, VivaFit oferă oportunitatea oamenilor specializați în nutriție și sport să se înregistreze cu un cont profesionist unde vor putea să își promoveze serviciile și să interacționeze cu posibilii clienți într-o comunitate dedicată.

Având în vedere aceste caracteristici, VivaFit reușește să combine tehnologia avansată, disponibilitatea pe diferite platforme și suplețea în sprijinirea utilizatorilor pentru atingerea obiectivelor personale, prin oferirea unei experiențe depline și adaptate pentru fiecare utilizator. Prin integrarea acestor detalii, VivaFit se poziționează ca o alternativă inventivă și competitivă în privința aplicațiilor pentru fitness și nutriție, acoperind atât necesitățile personale ale utilizatorilor, cât și pe cele ale specialiștilor în acest domeniu.

## MyFitnessPal

„MyFitnessPal” reprezintă una dintre cele mai populare platforme dedicate susținerii unui mod de viață sănătos și echilibrat, furnizând funcții folositoare pentru monitorizarea alimentației și a activităților sportive. Aplicația pune la dispoziție utilizatorilor posibilitatea de a își înregistra zilnic alimentele din cadrul meselor, de a urmări numărul de calorii consumate și distribuția acestora în funcție de macronutrienți și monitorizarea activității fizice. Pe lângă acestea, aplicația permite vizualizarea și introducerea greutății corporale, precum și a nivelului de hidratare, iar pentru o automatizare mai eficientă, utilizatorii au opțiunea de a își conecta un dispozitiv extern pentru contorizarea pașilor efectuați sau a caloriilor arse.

În vederea elaborării unei aplicații flexibile dar și ușor de utilizat, platforma pune la dispoziție o amplă mulțime de date, ce cuprinde alimente și rețete din diferite categorii, contribuind la trecerea rapidă a produselor consumate în aplicație. Mai mult, include opțiunea de scanare a codurilor de bare, facilitând și mai mult eficiența de căutare a alimentelor industriale, iar prin intermediul unei interfețe primitoare, utilizatorii își pot observa grafic înregistrările introduse.

|  |
| --- |
| Fig. 1 – Capturi de ecran din aplicația „MyFtinessPal” |

## Eat & Track

„Eat & Track” este o aplicație mobilă destinată utilizatorilor cu un interes dezvoltat în sfera monitorizării în detaliu a alimentației și a parametrilor nutriționali zilnici. Platforma permite utilizatorilor să își introducă rapid mesele consumate, oferind informații detaliate despre distribuția macronutrienților și a micronutrienților în cadrul aportului caloric zilnic, punându-se accent pe alimentele, preparatele și rețetele autohtone din România, disponibile în cadrul mulțimii de informații, adaptat la piața regională, din cadrul platformei.

În afara faptului că aplicația oferă posibilitatea de monitorizare a aportului caloric și a valorilor nutriționale, „Eat & Track” permite și înregistrarea altor tipuri de informații precum aportul de apă, activitățile fizice, greutatea corporală și a altor repere esențiale pentru un stil de viață sănătos. Principalul element prin care această aplicație se diferențiază de restul este datorită abordării axate pe personalizarea datelor pe care o aplică, utilizatorii având posibilitatea de a crea și folosi rețete sau alimente proprii, cu posibilitatea ajustării porții ingerate în funcție de gramajul real consumat.

|  |
| --- |
| Primii pași în monitorizarea alimentației cu Eat & Track: Ghid pentru  începători  Fig. 2 – Capturi de ecran din aplicația „Eat & Track” |

## Cronometer

„Cronometer” este o aplicație avansată dedicată utilizatorilor care își doresc un control riguros asupra alimentației și a stării generale de sănătate, care se evidențiază printr-o mulțime de informații nutriționale, asociate alimentelor, atent verificat pentru asigurarea unui grad de acuratețe ridicat al valorilor introduse. Precum și celelalte platforme, aplicația are funcționalități de monitorizare a alimentelor consumate, dar se diferențiază prin oferirea de detalii complexe precum cantitatea de vitamine, minerale și aminoacizi ce îi poate conține un produs.

De asemenea, aplicația pune la dispoziția utilizatorilor posibilitatea de a își înregistra și urmări altfel de parametrii de sănătate suplimentari cum ar fi colesterolul, glicemia și tensiunea arterială. În plus, platforma vine în ajutorul utilizatorilor care doresc să urmeze un anumit stil de dietă prin posibilitatea de a își personaliza obiectivele nutriționale și prin sugestiile de preparate ce pot fi consumate. Totodată, rapoartele detaliate și graficele interactive de progres pe termen scurt și lung oferă utilizatorilor o perspectivă clară asupra evoluției proprii, permițând reglarea strategiilor alimentare și a programului de exerciții fizice în funcție de rezultatele obținute și de obiectivele stabilite.

|  |
| --- |
| Why You Should Be Tracking Your Micronutrients With Spencer D'Amelio From  Cronometer – 20 Fit  Fig. 3 – Capturi de ecran din aplicația „Cronometer” |

# Tehnologii utilizate

Pentru implementarea aplicației VivaFit, am optat pentru folosirea unui portofoliu amplu de tehnologii moderne și robuste, asigurând astfel nivelul crescut de securitate, flexibilitate, accesibilitate și performanță al aplicației. Procesul de selecție a acestor tehnologii s-a bazat pe un studiu detaliat, orientat către atingerea obiectivelor propuse ale aplicației.

## Angular

Angular[[2]](#footnote-2) reprezintă un cadru de dezvoltare modern (framework) creat de Google și este utilizat într-un mare număr de aplicații web de tip pagină unică (Single Page Application - SPA). Aceste tipuri de aplicații facilitează o experiență modernă și rapidă prin actualizarea conținutului în cadrul aceleași pagini, fără reîncărcarea completă a interfeței grafice. După cum subliniază „*Jayaram & Kumar* ”(2), aplicațiile de tip pagină unică au ajuns să fie considerate un standard în dezvoltarea contemporană a aplicațiilor web, datorită interfețelor interactive și fluide.

De asemenea, Angular are la bază folosirea unei arhitecturi modulare prin folosirea componentelor individuale, a unui mecanism evoluat de rutare și a gestionării complexe a formularelor, asigurând astfel posibilitatea de a realiza optimizări performante în cadrul aplicațiilor extinse, așa cum menționează autorii studiului, care analizează performanța aplicațiilor extinse implementate cu Angular.

Printre numeroasele beneficii tehnologice și optimizări amintim:

* „Angular Ivy”, un nou concept introdus în Angular 9, care a eficientizat resursele folosite de aplicație pentru gestionarea componentelor și a șabloanelor, fiind un nou motor implicit de afișare care a redus considerabil dimensiunea fișierelor generate, rezultând într-o creștere de viteză a vizualizării componentelor.
* „Încărcarea întârziată” (lazy loading) reușește să îmbunătățească timpul de pornire al aplicației prin descărcarea modulelor aplicației doar la momentul folosirii lor.
* Mecanisme precum compilarea anticipată (Ahead-of-Time) și eliminarea codului neutilizat (tree shaking) duc la eficientizarea aplicației prin producerea codului executabil înainte de rulare și prin micșorarea dimensiunii finale a fișierelor.
* Folosirea nativă a bibliotecii „RxJS” (Reactive Extensions for JavaScript) contribuie la gestionarea elegantă a fluxurilor de date asincrone, fiind un aspect esențial pentru aplicațiile interactive.
* Suportul complet pentru implementarea aplicațiilor de tip „PWA” (Progressive Web Apps) facilitează includerea unor mecanisme de cache, notificări și funcționare în lipsa conexiunii la internet, îmbunătățind astfel accesibilitatea și reducând necesitatea permanentă de conexiune la internet.

Efectul acestor mecanisme și tehnologii implementate în cadrul platformei Angular este remarcabil, așa cum se precizează în concluziile studiului: „*Inovațiile Angular, precum Ivy, încărcarea întârziată și suportul pentru PWA, au îmbunătățit semnificativ performanța, scalabilitatea și experiența utilizatorilor în aplicațiile de tip pagină unică.*” (2).

## Tailwind CSS

Tailwind CSS[[3]](#footnote-3) este un sistem de dezvoltare modern pentru stilizarea aplicațiilor web, bazat pe folosirea de clase utilitare, simplificând procesul de creare a interfețelor vizualizate pe diferite tipuri și mărimi de ecrane. Un beneficiu considerabil îl alcătuiește ușurința în utilizare, deoarece „*generează nume de clase apropiate de scopul lor, eliminând necesitatea memorării lor*” (3), micșorând astfel dificultatea curbei de învățare.

Totodată, prin faptul că „*majoritatea stilurilor pot fi aplicate direct prin clase preexistente*” (3), se oferă posibilitatea aplicării rapide a stilurilor fără a scrie CSS manual. Mai mult decât atât, Tailwind simplifică crearea interfețelor adaptabile în funcție de mărimile dispozitivului prin faptul că oferă „*o abordare mai curată, cu clase orientate pe diferite categorii de mărime*” (3).

Nu în ultimul rând, Tailwind, prin oportunitatea de a reutiliza componente deja create, facilitează aplicarea principiului „DRY” (Don't repeat yourself), ducând astfel la optimizarea aplicației prin creșterea vitezei de încărcare a paginilor și prin menținerea unui cod curat.

## JSON Web Token (JWT)

JWT[[4]](#footnote-4) reprezintă un sistem de furnizare a informațiilor între două părți, utilizat deseori în sistemele de tip „SSO” (Single Sign-On), datorită certitudinii de garantare a securității cu care este acreditat. Conform lucrării, „*Json Web Token (JWT) este utilizat pentru a acorda autorizații, deoarece permite transmiterea de informații între două părți într-un mod securizat*” (4).

Folosirea acestui tip de autorizare pe scară largă este datorată faptului că „*JWT are capacitatea de a include atât entități (ex: roluri ale utilizatorilor), cât și timpi de expirare într-un singur token*” (4), prescurtând astfel numărul de cereri către server, dar și pentru că „*token-ul JWT este semnat digital folosind o cheie secretă, ceea ce asigură integritatea și autenticitatea lui*” (4).

Așa cum se poate observa în Fig. 4, mecanismul de autentificare și autorizare cu JWT operează în felul următor: utilizatorul trimite datele de autentificare către server, iar dacă acestea sunt cele corespunzătoare, serverul generează și trimite ca răspuns un token JWT. Cu acest token, utilizatorul va putea accesa resursele protejate până în momentul deconectării din aplicație, fără a mai primi datele de autorizare de fiecare dată.

|  |
| --- |
| Fig. 4 – Mecanismul de autorizare prin intermediul token-ului JWT |

## OAuth2 (Open Authorization)

Token-ul de tip OAuth2[[5]](#footnote-5) este utilizat frecvent pentru autentificarea cu platforme terțe, cum ar fi Google sau Facebook, fiind un token de acces transmis de furnizorul de identitate după ce utilizatorul s-a autentificat cu succes. Scopul lui este de a permite, în mod securizat, preluarea datelor specifice ale utilizatorului, cum ar fi numele, adresa de email sau fotografia de profil, fără a fi necesară verificarea directă a datelor de autentificare și transmiterea lor către aplicație. La fel precum JWT, token-ul OAuth2 are ca întrebuințare autorizarea accesului la resurse protejate, având o durată de valabilitate limitată, ceea ce garantează securitatea și reducerea riscului de acces neautorizat.

Această metodă ușurează procesul de autentificare externă și dezvoltă un sentiment de încredere utilizatorilor în ceea ce privește protecția datelor personale. În cadrul aplicației VivaFit, aceste informații sunt preluate, salvate și procesate intern pentru a genera un token JWT, utilizând numele utilizatorului și o parolă secretă, facilitând astfel o integrare precisă și performantă în mecanismul propriu de autentificare și autorizare.

## WebSocket

WebSocket-urile reprezintă o tehnologie bazată pe folosirea unei conexiuni deschise și fără întrerupere, care oferă posibilitatea schimbului de informații între client și server în timp real. Avantajul considerabil al acestei tehnologii îl constituie faptul că, în comparație cu modul tipic de comunicare pe internet, unde fiecare răspuns este precedat de o cerere separată, WebSocket-urile permit datelor să fie transmise și primite continuu, atât de la server către client, dar și invers, pe tot parcursul conexiunii. Având în vedere acest beneficiu, această tehnologie este implementată în aplicațiile care necesită ca informațiile să ajungă instant la utilizator, cum întâlnim în platformele de tip social media, unde conversațiile și notificările trebuie să fie actualizate în timp real.

În vederea asigurării compatibilității pe toate navigatoarele, inclusiv pe cele care nu conțin suport nativ pentru WebSocket-uri, VivaFit a integrat acest mecanism cu biblioteca „SockJS[[6]](#footnote-6)”, care asigură comunicarea funcțională indiferent de tehnologia disponibilă pe browser.

## Spring Boot

Spring Boot[[7]](#footnote-7) este unul dintre cele mai folosite framework-uri pentru dezvoltarea aplicațiilor moderne Java, fapt datorită modului simplu de configurare, dar și modularității și ecosistemului extins de care dă dovadă. În mod cert, „*Spring Boot conduce clasamentul framework-urilor pentru aplicații Java. 67% dintre dezvoltatorii Java îl folosesc, conform unui studiu JetBrains(2022)*” (5). Procentajul ridicat de alegere a acestui framework se poate explica atât prin modul ușor de creare a unui proiect nou, cât și prin suportul oferit de numeroasele biblioteci integrate, care facilitează crearea aplicațiilor de tip enterprise.

Productivitatea crescută reprezintă unul dintre beneficiile majore, care se regăsește în cadrul acestui framework. „*Spring Boot ne face mai productivi prin convenții inteligente. Elimină munca repetitivă de configurare, permițându-ne să ne concentrăm pe logica de afaceri.*” (5), iar prin intermediul Spring Initializr[[8]](#footnote-8), un proiect operațional poate fi creat într-o perioadă scurtă de timp, având deja configurate dependențele fundamentale și structura de bază, sprijinind astfel dezvoltarea unui prototip foarte rapid.

În plus, framework-ul reușește să reducă substanțial complexitatea în proiectele de mari anvergură prin promovarea organizării modulare a codului și prin faptul că „*fiecare componentă poate avea propria configurație, iar @ComponentScan asigură gestionarea automată a dependențelor*” (5). Totodată, framework-ul creează un ecosistem bogat prin faptul că „*oferă integrare externă cu tehnologii populare: baze de date, mesagerie (Kafka, RabbitMQ), securitate (OAuth2) și observabilitate (Micrometer)*” (5), ceea ce sprijină dezvoltatorii în conectarea rapidă și accesibilă a serviciilor esențiale pentru o aplicație modernă.

De asemenea, prin „*Dependecy Management-ul lui Spring Boot, care asigură compatibilitate între biblioteci, eliminând conflictele de versiuni*” (5), gestionarea dependențelor este realizată automatizat prin intermediul unui „BOM” (Bill of Materials), simplificând astfel întreținerea și actualizarea proiectului la ultimele versiuni disponibile. Mai mult decât atât, Spring Boot oferă flexibilitate totală pe partea de configurare, deoarece „*Configurația se poate face prin application.properties, YAML, variabile de mediu sau programatic. Profilele (dev, test, prod) permit adaptarea la diferite medii fără cod duplicat*” (5), ajutând la managementul aplicațiilor în diverse scenarii de rulare.

O altă componentă valoroasă este „JPA” (Java Persistence API), care reprezintă un șablon standardizat pentru gestionarea interacțiunii dintre aplicațiile Java și bazele de date relaționale. Principalul avantaj al folosirii „JPA”-ului constă în capacitatea automată de a genera structura bazei de date în funcție de clasele de entități definite în cod. Așadar, prin folosirea unor simple adnotări precum @Entity, @Table, @Column sau @Id, dezvoltatorul are oportunitatea de a descrie structura tabelelor direct în codul sursă, iar în funcție de setările aplicate, tabelele vor fi create sau actualizate corespunzător în baza de date, eliminând astfel necesitatea definirii manuale a schemelor „SQL”.

Un alt avantaj important al framework-ului Spring Boot îl reprezintă suportul nativ extins pentru securitate, oferit prin modulul Spring Security. Astfel, mecanismele de autentificare și autorizare, atât pe bază de token-uri JWT, OAuth2, cât și prin conectare prin nume de utilizator și parolă, permit implementarea rapidă a componentei de confirmarea identității și atribuirea de permisiuni. Spring Security vine cu posibilitatea activării protecțiilor implicite împotriva vulnerabilităților comune în aplicațiile web, precum „CSRF” (Cross-Site Request Forgery) și „XSS” (Cross-Site Scripting). Mai mult, prin folosirea Spring Data JPA sau JdbcTemplate cu interogări parametrizate, framework-ul oferă protecție totală împotriva atacurilor de tip SQL Injection.

Nu în ultimul rând, pentru folosirea comunicării asincrone și performantă, a fost folosită componenta Spring WebFlux, fiind un model reactiv din ecosistemul Spring. Deseori, este utilizat pentru implementarea și integrarea „SSE”-urilor (Server-Sent Events), care permit trimiterea de fragmente de mesaje live, de la server către client, fără ca utilizatorul să trebuiască să inițieze cereri repetate.

## Python

Python[[9]](#footnote-9) este printre cele mai folosite limbaje de programare în domeniul analizei datelor, dar și pentru implementarea algoritmilor de învățare automată sau pentru crearea diferitelor tipuri de inteligență artificială, fiind utilizat atât pentru sintaxa accesibilă, cât și pentru ecosistemul extins de biblioteci specializate, oferind un suport minunat pentru implementarea rapidă a programelor. În cadrul aplicației dezvoltate, Python a jucat un rol important în implementarea unor funcționalități precum comunicația cu dispozitive Bluetooth, dar și pentru prelucrarea și datelor, în scopul de a anticipa anumite valori.

Biblioteca Bleak a avut ca scop detectarea dispozitivelor care au pornită funcția de Bluetooth din jurul echipamentului de pe care rulează aplicația, dar și primirea informațiilor transmise prin Bluetooth Low Energy (BLE) de la acestea, iar în combinație cu biblioteca asyncio, a fost posibilă gestionarea asincronă a conexiunilor cu dispozitive externe.

Pentru implementarea funcționalităților de învățare automată și inteligență artificială, aplicația integrează două componente specializate:

* Un algoritm de machine learning, antrenat cu ajutorul bibliotecii scikit-learn, pentru a estima și analiza trendul greutății corporale pe baza istoricului înregistrat de utilizator.
* O rețea neuronală artificială, dezvoltată prin intermediul bibliotecii TensorFlow Keras, utilizată în scopul de a prezice numărul de calorii arse, în timpul activităților fizice, în funcție de anumite date adăugate.

Această abordare a facilitat integrarea comunicației hardware cu procese predictive, sprijinind astfel personalizarea recomandărilor și monitorizarea parametrilor corporali.

## SendGrid API

SendGrid[[10]](#footnote-10), parte din ecosistemul Twilio, reprezintă o platformă fiabilă pentru livrarea email-urilor, fiind utilizată în cadrul multor aplicații pentru că oferă dezvoltatorilor și echipelor de produs posibilitatea de a își optimiza comunicările prin email, având la dispoziție o infrastructură dedicată și unelte specializate, pentru a fi sigură livrarea eficientă a mesajelor direct în inbox-ul destinatarilor.

Folosirea acestui serviciu aduce beneficii majore în cadrul aplicației dezvoltate precum: rată de livrare garantată a email-urilor, un nivel crescut de securitate avansată prin folosirea protocoalelor de autentificare specifice trimiterii de email-uri, posibilitatea de personalizare a conținutului și de gestionare a șabloanelor, dar și integrarea rapidă prin intermediul API-urilor moderne și bine documentate.

În cadrul aplicației dezvoltate, utilizarea serviciului SendGrid a fost realizată pentru integrarea a trei aspecte: trimiterea codurilor de verificare pentru procesul de autentificare, livrarea link-urilor de resetare a parolei și pentru crearea de alerte și notificări în cazul conectării de pe dispozitive necunoscute.

Așadar, această integrare a dus la creșterea atât a nivelului de securitate, cât și a calității experienței utilizatorilor, facilitând astfel o comunicare rapidă, sigură și controlată.

## Ollama

Ollama[[11]](#footnote-11) este o opțiune modernă și accesibilă pentru integrarea modelelor de limbaj de tip „LLM” (Large Language Model) în aplicații, având ca posibilitate accesarea modelelor conversaționale atât local, cât și în cloud.

Pentru aplicația dezvoltată, am ales utilizarea modelului aya-expense, deoarece s-a evidențiat prin procesarea eficientă a limbii române și prin faptul că înțelege inflexiunile conversaționale legate de domeniul nutriției și fitness-ului. Un prim rol pe care l-a avut modelul a fost utilizarea în cadrul discuțiilor interactive de tip chat cu utilizatorii, susținând așadar schimbul de informații asociate stilului de viață sănătos, planurilor de alimentație și recomandărilor de activități fizice.

Dincolo de folosirea pentru implementarea componentei conversaționale, modelul a fost aplicat și pentru analiza compoziției meselor și a obiceiurilor alimentare ale utilizatorilor, producând astfel sugestii personalizate în funcție de obiectivele și istoricul fiecăruia.

În cadrul aplicației, integrarea acestui tip de tehnologie și a modelului ales au furnizat avantaje semnificative spre exemplu suportul pentru limba română precis în conversații și procesarea textului, implementarea flexibilă și calitativă prin API, posibilitatea de a selecta modele pentru domenii specifice și oportunitatea rulării locale a modelelor, fără a fi dependent de infrastructuri externe costisitoare.

Așadar, Ollama devine o variantă performantă pentru particularizarea interacțiunii cu utilizatorii și pentru creșterea calității recomandărilor oferite.

## MariaDB

MariaDB[[12]](#footnote-12) reprezintă o alternativă îmbunătățită și compatibilă binar pentru MySQL, aceasta fiind creată pe fondul nemulțumirilor legate de ritmul lent de dezvoltare și calitatea slabă a versiunilor lansate după achiziția MySQL de către Sun: „*MariaDB este un înlocuitor îmbunătățit, compatibil binar, pentru MySQL, creat de câțiva dintre foștii dezvoltatori principali ai MySQL*” (6).

MariaDB oferă anumite avantaje importante față de MySQL. De exemplu, „*MariaDB oferă un mecanism prin care clienții pot primi mesaje de progres de la server*” (6), prevenind astfel întreruperea prematură a comenzilor lungi, cu o complexitate ridicată. Mai mult decât atât, performanța generală a fost și optimizată: „*Optimizatorul îmbunătățit din MariaDB funcționează semnificativ mai rapid decât MySQL pentru sarcini complexe*” (6). Un alt element ce contribuie la consolidarea bazei de date MariaDB este implementarea politicii stricte de testare și remediere a eventualelor erori, rezultând într-o „*reducere semnificativă a numărului total de bug-uri și a avertismentelor de compilator*” (6).

În cadrul aplicației dezvoltate, sistemul de gestiune a tabelelor din baza de date a fost realizată de MariaDB, datorită fiabilității, compatibilității remarcabile cu MySQL și a performanței îmbunătățite. Această bază de date a fost utilizată pentru salvarea și administrarea mai multor categorii de informații fundamentale pentru funcționarea corectă a aplicației, precum: datele de autentificare ale utilizatorilor, valorile calculate pentru BMR (Basal Metabolic Rate), alimentele, rețetele, preparatele consumate la fiecare masă și alte date legate de monitorizarea stilului de viață. Această alegere a facilitat o organizare performantă a informațiilor prin rapiditatea interogărilor și prin siguranța datelor utilizatorului.

Prin urmare, „*Cu MariaDB, nu doar că obții toate lucrurile bune din MySQL, dar mai beneficiezi și de funcții suplimentare, îmbunătățiri de performanță, teste mai bune și mai puține bug-uri. Pentru un număr tot mai mare de persoane și companii, alegerea este clară: MariaDB este viitorul MySQL*” (6).

# Arhitectura aplicației

În acest capitol, urmează să documentez, în mod structurat, arhitectura aplicației dezvoltate, care a fost realizată prin intermediul modelului C4, propus de Simon Brown. Această metodă oferă posibilitatea de a descrie gradual un sistem software, plecând de la o vedere de ansamblu asupra contextului în care funcționează și ajungând până la detalii fine privind implementarea și organizarea codului.

În cadrul acestei părți de lucrare, arhitectura va fi prezentată în cinci subcapitole diferite:

* Nivelul 1 – Diagrama de Context: prezintă o imagine colectivă a modului în care aplicația interacționează atât cu utilizatorii, cât și cu serviciile externe pe care le folosește.
* Nivelul 2 – Diagrama de Containere: pune în detaliu modul în care aplicația este divizată în componente mari (containere) și modul de funcționare între acestea.
* Nivelul 3 – Diagrama de Componente: oferă o imagine asupra structurii interne a fiecărui container, punând în evidență principalele componente software și modul de relaționare între ele.
* Nivelul 4 – Diagrama de Cod: ilustrează detalii fine la nivelul claselor (entităților), precum și funcționalitatea acesteia în cadrul unei componente.
* Arhitectura bazei de date: prezintă structură logică a tabelelor din cadrul bazei de date, oferind informații despre relațiile dintre ele și modul de administrare a datelor.

Printr-o astfel de abordare progresivă, se crește rata de înțelegere a arhitecturii generale, precum și a detaliilor tehnice importante, punând la dispoziție o documentație clară și coerentă a aplicației.

## Diagrama C4

### Nivelul 1: Context

Diagrama de context reprezintă primul nivel al modelului C4, care prezintă o vedere de ansamblu asupra sistemului VivaFit, precum și a actorilor principali care interacționează cu acesta. Crearea acestei diagrame are drept scop înțelegerea limitelor aplicației, a fluxurilor de date și a responsabilității fiecărei părți implicate.

|  |
| --- |
| Fig. 5 – Diagrama C4: Nivel 1 |

După cum se poate observa în Fig. 5, actorii aplicației sunt „Utilizator”, „Specialist” și „Admin”, iar sistemele externe integrate sunt „SendGrid”, „Ollama” și „Autorizare Google/Facebook”. În ceea ce privește fluxurile principale, putem distinge faptul că utilizatorii folosesc VivaFit pentru a își gestiona sănătatea, specialiștii își promovează serviciile pe platformă, iar aplicația se folosește de servicii externe pentru a integra funcționalități avansate.

### Nivelul 2: Containere

Nivelul 2 al modelului C4 oferă detalii despre containerele principale ale aplicației VivaFit, precum componente cheie care alcătuiesc sistemul și modul în care acestea interacționează. În cadrul sistemului VivaFit, un container poate fi o aplicație sau o bază de date.

|  |
| --- |
| Fig. 6 – Diagrama C4: Nivel 2 |

După cum se poate vedea din Fig. 6, aplicația VivaFit este formată din patru containere, fiecare având rolul și funcționalitățile sale. Spre exemplu, „Aplicația Web” folosește tehnologii precum Angular, Tailwind, CSS și are drept scop furnizarea interfeței utilizatorului și a conținutului dinamic. „Aplicația Single-Page” permite utilizatorilor să interacționeze cu VivaFit prin intermediul unei interfețe implementate cu Angular, Tailwind și CSS. „Aplicația RESTful API” utilizează SpringBoot, Python și TypeScript pentru a apela și a administra endpoint-urile aplicației, punând spre expunere funcționalitățile aplicației, iar „Baza de date” folosește ca tehnologie MariaDB și stochează informații precum credențialele utilizatorilor, mesele și alimentele introduse și alte detalii nutriționale.

### Nivelul 3: Componente

Nivelul 3 al modelului C4 descrie în detaliu componentele interne ale fiecărui container din aplicația VivaFit, explicând funcționalitățile specifice și interacțiunile dintre ele. Astfel, acestea reprezintă unitățile funcționale care implementează logica de bază a aplicației.

|  |
| --- |
| Fig. 7 – Diagrama C4: Nivel 3 |

Așa cum se poate observa și în Fig. 7, componentele cheie ale aplicației VivaFit garantează nivelul complet de funcționare pentru gestionarea unui stil de viață echilibrat și facilitează interacțiunea dintre diferitele tipuri de utilizatori. Modulul de gestionare a utilizatorilor (Autentificare, Înregistrare, Profil) se ocupă de securitate și personalizare, în timp ce monitorizarea nutriției, exercițiilor fizice și a obiectivelor se realizează prin modulul de monitorizare a stilului de viață (BMR, Jurnal, Rapoarte, Rețete). Specialiștii beneficiază de instrumente dedicate (Specialist), iar utilizatorii pot interacționa cu aceștia prin intermediul componentei „Specialiști”. În plus, componenta „AI” extinde capabilitățile aplicației atât prin analize și recomandări avansate, cât și prin funcționalitatea de a susține conversații interactive despre nutriție și fitness.

### Nivelul 4: Cod

Nivelul 4, din cadrul modelului C4, reprezintă implementarea concretă a aplicației, descriind astfel concepte precum clasele individuale și atributele lor, relațiile dintre entități, dar și logica de business. Acesta reprezintă nivelul cel mai detaliat, fiind echivalent cu diagramele UML de clasă sau cu structura tabelelor din baza de date.

|  |
| --- |
| Fig. 8 – Diagrama C4: Nivel 4 - Autentificarea |

În conformitate cu Fig. 8, Sistemul de autentificare gestionează toate aspectele legate de managementul utilizatorilor și securitate, fiind format din entități precum:

* User – memorează informațiile de bază ale utilizatorilor (email, nume de utilizator, parolă criptată, rol, număr de telefon).
* PendingUsers & PendingSignInUser – gestionează înregistrările și procesele de autentificare în așteptare, în cazul necesității validării prin coduri de confirmare trimise pe email.
* ConfirmationCode – stochează codurile de confirmare generate în urma solicitărilor de înregistrare sau autentificare (în cazul autentificării de pe rețele sau dispozitive noi).
* PasswordResetToken – asigură securitatea operațiuni temporare de a reseta parola.
* SignInToken – verifică credibilitatea token-urilor JWT trimise prin intermediul mecanismului de autorizare, în cadrul cererilor API.
* ConnectionDetails – monitorizează dispozitivele și rețelele salvate de către utilizator.

|  |
| --- |
| Fig. 9 – Diagrama C4: Nivel 4 – Gestionarea caloriilor |

Diagrama, prezentată în Fig. 9, detaliază structura de date pentru gestionarea alimentației și a caloriilor în aplicația VivaFit. În cadrul acestui modul, entitățile regăsite sunt: Meal (reprezintă o masă care are o dată și tipul mesei asociate), Food (stochează informațiile despre alimente), MealFood (asociază alimente cu mese și include valorile nutriționale), Recipe (conține informațiile necesare gătirii unei rețete), FoodFavorite (alimentele favorite ale utilizatorilor), BMRDetails (calculează și stochează necesarul zilnic de calorii și alte detalii fizice ale utilizatorilor).

|  |
| --- |
| Fig. 10 – Diagrama C4: Nivel 4 – Gestionarea exercițiilor și a greutății |

Modulul, prezentat în Fig. 10, are drept scop administrarea greutății corporale și a activităților fizice înregistrate de către utilizator. Acesta conține următoarele entități: Weight (permite înregistrarea masei corporale și urmărirea evoluției greutății utilizatorului), Exercise (definește exercițiile standard prin descriere, categorie și alte detalii) și UserExercise (înregistrează activitățile fizice personalizate ale fiecărui utilizator).

|  |
| --- |
| Fig. 11 – Diagrama C4: Nivel 4 – Gestionarea specialiștilor |

Modulul de management al specialiștilor și al mesajelor acestora, detaliat în Fig. 11, permite antrenorilor și nutriționiștilor să își gestioneze profilul și interacțiunile cu potențialii clienți. Această parte a aplicației implementează următoarele entități: Specialist (detaliile complete ale specialistului precum profil social, experiență, programe oferite), MessageSpecialist (mesajele primite de la utilizatori), User (specialiștii sunt și ei utilizatori cu rol diferit, iar utilizatorii obișnuiți pot trimite mesaje către specialiști).

Având în vedere diagramele prezentate din cadrul nivelului 4, al modelului C4, structura implementată oferă modularitate și flexibilitate prin separarea clară a componentelor, scalabilitate, performanță la nivelul interogărilor și personalizare prin intermediul diferitelor roluri integrate.

## Arhitectura bazei de date

Baza de date a aplicației VivaFit constituie fundamentul tehnologic care susține toate funcționalitățile platformei, fiind construită să gestioneze eficient diversele date complexe ale utilizatorilor. Structura sa implementează o abordare generală pentru monitorizarea sănătății, îmbinând teme asociate nutriției și fitness-ului într-un ecosistem coerent.

Precum am detaliat în Fig. 12 (vezi mai jos), sistemul de salvare a informațiilor este organizat în cinci domenii funcționale principale:

* Managementul utilizatorilor – include tabele pentru administrarea conturilor (users), procesele de autentificare și autorizare (password\_reset\_token, sign\_in\_tokens) și datele temporare ale utilizatorilor aflați în proces de validare (pending\_sign\_up\_users, pending\_sign\_in\_users).
* Calculul personalizat al sănătății – centrat pe tabela „bmr\_details” care salvează parametrii individuali fizici și calculează necesitățile nutriționale.
* Gestionarea alimentației – este structurat pe cinci niveluri: alimentele de bază (foods), rețetele mai complexe (recipes), mesele compuse (meals), asocierea dintre alimente și mese (meal\_foods), precum și mecanismul de salvare a preferințelor utilizatorilor (food\_favorites).
* Monitorizarea activităților fizice – conține atât exercițiile standard (exercise), cât și activitățile personale ale utilizatorilor, dar permite și urmărirea progresului prin tabela dedicată greutății (weights).
* Managementul specialiștilor – salvează informațiile pentru profilul profesional al specialiștilor (specialists – extins din user cu detalii profesionale) și gestionează primirea și trimiterea mesajelor către specialiști.

În ceea ce privește relațiile și interconexiunile, arhitectura utilizează inteligent relațiile relaționale pentru a asigura integritatea datelor. Spre exemplu, relațiile one-to-many sunt utilizate între utilizatori și activitățile lor, iar relațiile many-to-many sunt folosite prin tabele de legătură specializate (meal\_foods pentru asocierea alimentelor cu mesele).

Modul în care este implementată baza de date aduce beneficii competitive precum performanța optimizată, adaptabilitate și scalabilitate datorită designului modular, securitate înaltă prin separarea clară între datele sensibile și cele operaționale.

|  |
| --- |
| Fig. 12 – Arhitectura bazei de date |

Pentru a încheia, arhitectura bazei de date a aplicației VivaFit a fost concepută astfel încât să ofere un cadru coerent și bine organizat, capabil să susțină toate funcționalitățile esențiale ale platformei. Structura logică a fost gândită pentru a permite o administrare eficientă a informațiilor, cu relații bine definite între componente, astfel încât datele utilizatorilor și interacțiunile dintre aceștia să fie gestionate corect și sigur. Pe de o parte, această schemă răspunde cerințelor actuale ale aplicației, iar pe de altă parte, oferă o fundație solidă pentru extinderi viitoare, adaptându-se ușor la noi funcționalități sau provocări din domeniu. Prin arhitectura modulară și relațională implementată, VivaFit își poate susține dezvoltarea pe termen lung, menținând în același timp un nivel ridicat de performanță, chiar și în condițiile unui număr tot mai mare de utilizatori, consolidându-și astfel poziția în zona sănătății digitale.

1. Plurivalent: capacitatea de a avea mai multe interpretări, folosită pentru a exprima complexitatea unui concept. [↑](#footnote-ref-1)
2. https://angular.dev/ [↑](#footnote-ref-2)
3. https://tailwindcss.com/ [↑](#footnote-ref-3)
4. https://jwt.io/ [↑](#footnote-ref-4)
5. https://www.oauth.com/oauth2-servers/access-tokens/ [↑](#footnote-ref-5)
6. https://github.com/sockjs/sockjs-client [↑](#footnote-ref-6)
7. https://spring.io/projects/spring-boot [↑](#footnote-ref-7)
8. https://start.spring.io/ [↑](#footnote-ref-8)
9. https://www.python.org/ [↑](#footnote-ref-9)
10. https://sendgrid.com/ [↑](#footnote-ref-10)
11. https://ollama.com/ [↑](#footnote-ref-11)
12. https://mariadb.org/ [↑](#footnote-ref-12)