

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare și Tehnologia Informației
SPECIALIZAREA: Tehnologia Informației

LUCRARE DE DIPLOMĂ

Coordonator științific:
Conf. dr. ing. Simona CARAIMAN

Absolvent:
Ștefan
FUNDUEANU-CONSTANTIN

Iași, 2024

UNIVERSITATEA TEHNICĂ „Gheorghe Asachi” din IAȘI
FACULTATEA DE AUTOMATICĂ ȘI CALCULATOARE
DOMENIUL: Calculatoare și Tehnologia Informației
SPECIALIZAREA: Tehnologia Informației

Estimarea compoziției corporale utilizând algoritmi de învățare automată

LUCRARE DE DIPLOMĂ

Coordonator științific:
Conf. dr. ing. Simona CARAIMAN

Absolvent:
Ștefan
FUNDUEANU-CONSTANTIN

Iași, 2024

**DECLARAȚIE DE ASUMARE A AUTENTICITĂȚII
LUCRĂRII DE DIPLOMĂ**

Subsemnatul(a) ȚUNDEANU-CONSTANTIN ȘTEFAN,
legitimat(ă) cu CI seria 147 nr. 789345, CNP 5010503226770
autorul lucrării ESTIMAREA COMPOZIȚIEI CORPORALE
UTILIZÂND ALGORITMI DE ÎNVĂȚARE AUTOMATĂ

elaborată în vederea susținerii examenului de finalizare a studiilor de licență organizat de către Facultatea de Automatică și Calculatoare din cadrul Universității Tehnice „Gheorghe Asachi” din Iași, sesiunea Iulie a anului universitar 2023-2024, luând în considerare conținutul Art. 34 din Codul de etică universitară al Universității Tehnice „Gheorghe Asachi” din Iași (Manualul Procedurilor, UTI.POM.02 – Funcționarea Comisiei de etică universitară), declar pe proprie răspundere, că această lucrare este rezultatul propriei activități intelectuale, nu conține porțiuni plagiate, iar sursele bibliografice au fost folosite cu respectarea legislației române (legea 8/1996) și a convențiilor internaționale privind drepturile de autor.

Data
01.07.2024

Semnătura


Cuprins

Introducere	1
1 Fundamentarea teoretică și documentarea bibliografică	3
1.1 Domeniul și contextul abordării temei	3
1.2 Tema propusă	4
1.2.1 Obiective și justificarea abordării	4
1.3 Realizările actuale pe aceeași temă și analiza tipurilor de produse/ aplicații existente	4
1.3.1 Provocări etice și beneficii ale tehnologiei mHealth	9
1.4 Elaborarea specificațiilor privind caracteristicile așteptate de la aplicație	9
1.4.1 Tehnologii folosite pentru implementare	9
1.4.2 Caracteristicile așteptate de la aplicație	10
1.5 Tendințe și inovații în aplicațiile de sănătate și fitness	11
1.5.1 Inteligența artificială și învățarea automată	11
1.5.2 Internetul lucrurilor (IoT)	11
1.5.3 Realitatea augmentată și virtuală	11
1.5.4 Monitorizarea sănătății în timp real	11
1.5.5 Personalizarea experienței utilizatorilor	11
1.6 Concluzii	12
2 Proiectarea aplicației	13
2.1 Concepte generale	13
2.1.1 Obiectivele aplicației	13
2.2 Arhitectura aplicației	14
2.2.1 Analiza platformei hardware	14
2.2.2 Analiza platformei software	14
2.2.3 Avantaje	14
2.2.4 Limitări și dezavantaje	15
2.3 Achiziția și salvarea datelor	15
2.3.1 Descriere generală	15
2.3.2 Setul de date utilizat	16
2.3.3 Variabilele setului de date	16
2.3.4 Prelucrarea și analiza datelor	16
2.3.5 Integrarea cu Firebase	16
2.3.6 Monitorizarea activităților fizice	16
2.3.7 API-ul pentru știri	17
2.3.8 API-ul Google pentru locație și harta sălilor	17
2.4 Componenta software	17
2.4.1 Arhitectura software a aplicației	17
2.4.2 Diagrame UML	18
2.4.3 Modulele generale ale aplicației și interacțiunile dintre ele	19

2.4.4	Gestionarea autentificării utilizatorilor/Interacțiunea cu Firebase	20
2.4.4.1	Descriere generală	20
2.4.4.2	Înregistrarea utilizatorilor	20
2.4.4.3	Autentificarea utilizatorilor	21
2.4.4.4	Resetarea parolei	21
2.4.4.5	Actualizarea datelor utilizatorului	21
2.4.4.6	Gestionarea datelor în Firebase Realtime Database	21
2.4.5	Limitările utilizării serviciilor Firebase	21
2.5	Modulul de estimare a procentajului de grăsime	22
2.5.1	Avantajele Metodei OLS și Regresiei Liniare	22
2.5.2	Procesul de Curățare a Datelor	22
2.5.2.1	Identificarea și eliminarea outlier-ilor	22
2.5.2.2	Standardizarea datelor	22
2.5.3	Evaluarea modelului	22
2.5.4	Limitările metodei alese	23
2.5.5	Compararea modelului cu alte tehnici	24
2.5.6	Concluzii și avantaje ale abordării propuse	25
2.6	Analiza performanțelor și descrierea componentelor hardware	25
2.6.1	Performanța aplicației pe diferite dispozitive	25
2.6.2	Descrierea componentelor hardware	25
2.7	Concluzii și direcții viitoare	26
3	Implementarea aplicației	27
3.1	Descrierea generală a implementării	27
3.2	Probleme speciale/dificultăți întâmpinate și modalități de rezolvare	27
3.2.1	Probleme de integrare a Firebase	27
3.2.2	Calculul și afișarea BMI și a grăsimii corporale	28
3.2.3	Localizarea sălilor de fitness	28
3.2.4	Performanța modelului de învățare automată	28
3.2.5	Probleme de compatibilitate a dispozitivelor	28
3.2.6	Contorizarea pașilor	29
3.3	Idei originale, soluții noi	29
3.3.1	Utilizarea algoritmilor de învățare automată pentru estimarea compoziției corporale	29
3.3.2	Monitorizarea activității fizice prin senzori	29
3.3.3	Calculul necesarului caloric și integrarea NewsAPI	29
3.4	Funcționarea sistemului	29
3.5	Comunicarea cu alte sisteme și salvarea/ stocarea informațiilor	36
3.6	Interfața cu utilizatorul	37
3.6.1	Integrarea componentelor UI	38
4	Testarea aplicației și rezultate experimentale	39
4.1	Punerea în funcțiune/ lansarea aplicației, elemente de configurare sau instalare . . .	39
4.2	Testarea sistemului (hardware/ software)	39
4.3	Aspecte legate de încărcarea procesorului, memoriei, limitări în ce privește transmisia datelor/ comunicarea	40
4.3.1	Utilizarea CPU	40
4.3.2	Utilizarea Memoriei	40
4.4	Date de test/ metrice/ benchmarks	41
4.5	Aspecte legate de fiabilitate/ securitate	42

4.6	Rezultate experimentale	43
Concluzii		47
4.7	Evidențierea concisă a contribuțiilor/ soluțiilor personale	47
Bibliografie		51

Estimarea compoziției corporale utilizând algoritmi de învățare automată

Ștefan FUNDUEANU-CONSTANTIN

Rezumat

Această lucrare de licență se concentrează pe dezvoltarea unei aplicații mobile Android care utilizează algoritmi de învățare automată pentru estimarea compoziției corporale. Obiectivele principale includ crearea unui model de regresie liniară pentru estimarea procentului de grăsime corporală pe baza unor măsurători antropometrice simple și integrarea acestui model într-o aplicație mobilă prietenoasă pentru utilizator.

Progresele tehnologice moderne au permis colectarea în timp real a datelor biometrice prin intermediul dispozitivelor mobile, făcând monitorizarea personalizată a sănătății și a condiției fizice mai accesibilă. Acest proiect valorifică aceste capacități pentru a oferi utilizatorilor o metodă convenabilă de monitorizare a compoziției corporale care, tradițional necesită echipamente costisitoare și specializate.

Metodologia utilizată în această lucrare implică dezvoltarea și evaluarea modelului de învățare automată folosind bibliotecile Scikit-learn și Pandas, iar Firebase este utilizat pentru autentificare și stocarea datelor. Aplicația mobilă este dezvoltată în Java, asigurând stabilitatea și compatibilitatea cu platforma Android. Funcționalitățile principale ale aplicației includ autentificarea utilizatorilor și gestionarea profilului acestora, introducerea și stocarea datelor antropometrice, estimarea procentului de grăsime corporală, monitorizarea pașilor zilnici, calculul necesarului caloric zilnic și calcularea BMI-ului (Body Mass Index).

Prin utilizarea senzorilor dispozitivului mobil, aplicația poate monitoriza pașii zilnici efectuați de către utilizatori, stocând și sincronizând datele în Firebase Realtime Database, asigurând astfel accesul facil la istoricul activităților. De asemenea, aplicația calculează necesarul caloric zilnic pe baza obiectivelor setate și a nivelului de activitate fizică, oferind utilizatorilor informații valoroase și personalizate pentru gestionarea sănătății și condiției fizice.

Aplicațiile actuale de sănătate și fitness oferă funcționalități similare, dar deseori depind de hardware suplimentar și pot fi costisitoare. Aplicația noastră își propune să ofere o alternativă accesibilă, utilizând doar smartphone-ul utilizatorului.

Rezultatele obținute demonstrează potențialul aplicației de a contribui la sănătatea publică, deoarece oferă un instrument precis și ușor de utilizat pentru monitorizarea compoziției corporale și a altor parametri de sănătate, sprijinind utilizatorii în menținerea unui stil de viață sănătos. Pe lângă funcționalitățile principale, aplicația se integrează și cu alte servicii, precum Google Maps API, pentru a oferi informații relevante și actualizate despre locațiile sălilor de sport. Aceste caracteristici inovatoare, împreună cu designul intuitiv și accesibil, asigură o experiență de utilizare prietenoasă și eficientă.

Introducere

Motivația alegerii temei

În zilele noastre, pentru mulți oameni, sănătatea și bunăstarea personală sunt prioritare. Datorită progreselor tehnologice, utilizarea smartphone-urilor pentru monitorizarea diferitelor aspecte ale sănătății, cum ar fi compoziția corporală, nivelul de activitate fizic și necesarul caloric s-a răspândit enorm. Motivația alegerii temei printr-o abordare bazată pe algoritmi de învățare automată constă în intenția de a oferi o modalitate accesibilă și exactă utilizatorilor pentru monitorizarea stării lor de sănătate, fără a necesita echipamente costisitoare sau specializate. Utilizatorii pot obține informații detaliate și personalizate despre starea lor de sănătate prin intermediul acestei aplicații mobile, ceea ce îi va ajuta să adopte un stil de viață sănătos.

Relevanța și contextul temei alese

În ultima vreme, domeniul sănătății mobile, denumit și mHealth, a avut o dezvoltare considerabilă. Motivul acestui fenomen constă în creșterea accesibilității la smartphone-uri și progresul dezvoltării senzorilor avansați încorporați în aceste dispozitive. Aplicațiile mobile de sănătate nu se limitează doar la monitorizarea parametrilor de sănătate ai utilizatorilor, ci oferă și recomandări personalizate pentru a îmbunătăți stilul de viață, prevenirea bolilor și gestionarea afecțiunilor cronice. Având în vedere această situație, este indispensabilă dezvoltarea de noi idei, produse astfel încât să satisfacem cerințele unei audiențe tot mai conștiente de importanța sănătății personale.

Obiective generale ale lucrării

Această lucrare are în vedere dezvoltarea unei aplicații mobile Android ce utilizează algoritmi de învățare automată pentru a estima compoziția corporală a utilizatorilor ca scop principal. Obiectivele specifice includ:

- Se poate crea un model de regresie liniară pentru a estima procentul de grăsime corporală utilizând măsuratori antropometrice simple.
- Incorporarea modelului de învățare automată într-o aplicație mobilă Android.
- Testarea performanței modelului și a aplicației într-un mediu real de utilizare.
- Furnizarea către utilizatori a unei metode precise și ușor accesibile pentru monitorizarea compoziției corporale și a altor indicatori de sănătate.

Metodologia și instrumentele utilizate

Metodologia adoptată în această lucrare implică utilizarea de tehnologii moderne și practici bine stabilite în dezvoltarea software. Bibliotecile Scikit-learn și Pandas au fost utilizate în mediul de dezvoltare Google Colab pentru a dezvolta modelul de învățare automată. Am antrenat și evaluat modelul de regresie liniară pe un set de date colectat anterior pentru a măsura performanța sa folosind metriche precum Mean Absolute Error (MAE), Mean Squared Error (MSE) și coeficientul de determinare (R^2).

Pentru dezvoltarea aplicației mobile s-a utilizat limbajul de programare Java și Android SDK. Pentru a gestiona autentificarea și stocarea datelor, s-au utilizat Firebase Authentication și Firebase Realtime Database ca servicii. În plus, aplicația utilizează senzorii dispozitivului mobil pentru a urmări activitatea fizică a utilizatorilor.

Structura lucrării

Lucrarea este structurată în mai multe capitole, fiecare abordând un aspect specific al dezvoltării aplicației:

- **Capitolul 1: Fundamentarea teoretică și documentarea bibliografică pentru tema propusă.** În acest capitol, se va aborda contextul și importanța temei, împreună cu o evaluare a progreselor actuale în aceeași direcție.
- **Capitolul 2: Proiectarea sistemului.** În acest capitol sunt explicate detaliat arhitectura aplicației, modulele componente și modul lor de interacțiune.
- **Capitolul 3: Implementarea aplicației.** În această parte se expun în detaliu etapele de dezvoltare a aplicației, evidențiind atât dificultățile întâmpinate, cât și soluțiile adoptate pentru acestea.
- **Capitolul 4: Efectuarea testelor pentru aplicație și obținerea rezultatelor experimentale.** Acest capitol prezintă metodele de testare, metricile utilizate și rezultatele obținute.
- **Capitolul 5: Rezultate și perspective în continuare.** În acest capitol se face o sinteză a lucrărilor realizate și se propun posibile direcții de dezvoltare în continuare a aplicației.

Capitolul 1. Fundamentarea teoretică și documentarea bibliografică

1.1. Domeniul și contextul abordării temei

Tema lucrării de licență se încadrează în domeniul **inteligenței artificiale (AI)**, cu un focus specific pe **aplicațiile mobile pentru sănătate digitală și wellness-ul personal**. În ultimii ani, tehnologia mobilă a devenit din ce în ce mai integrată în viața de zi cu zi, iar utilizarea **smartphone-urilor pentru monitorizarea sănătății și fitness-ului** a crescut semnificativ. Acest context oferă oportunități noi pentru dezvoltarea de soluții inovatoare care să ajute utilizatorii să-și gestioneze sănătatea și să-și îmbunătățească stilul de viață [1].

Datorită progreselor în domeniul senzorilor mobili și al tehnologiilor de comunicare, colectarea de date biometrice în timp real este acum posibilă cu dispozitivele mobile. Aceste date sunt apoi analizate folosind algoritmi de învățare care oferă automat informații relevante și personalizate despre starea de sănătate a utilizatorilor. Prin urmare, aplicațiile mobile pentru sănătate nu doar urmăresc parametrii de sănătate a oamenilor, ci și oferă sfaturi despre cum să-și îmbunătățească stilul de viață, să prevină bolile și să gestioneze afecțiunile cronice.

Tehnologiile mobile actuale combină capacitatea mare de procesare a smartphone-urilor cu capacitatea lor de a colecta și transmite date în timp real. Acestea includ, dar nu se limitează la, senzori GPS, accelerometre, giroscopice și conectivitate de mare viteză la internet. Combinația acestor tehnologii permite colectarea unei game variate de date biometrice și comportamentale.

O creștere a interesului pentru sănătatea personală și fitness, împreună cu accesul mai bun la tehnologie, a condus la o cerere mai mare de produse digitale personalizate (Figura 1.1). Aceasta este alimentată, de asemenea, de tendințele globale de urbanizare și stilurile de viață sedentare care necesită intervenții rapide și eficiente pentru prevenirea și gestionarea bolilor cronice.

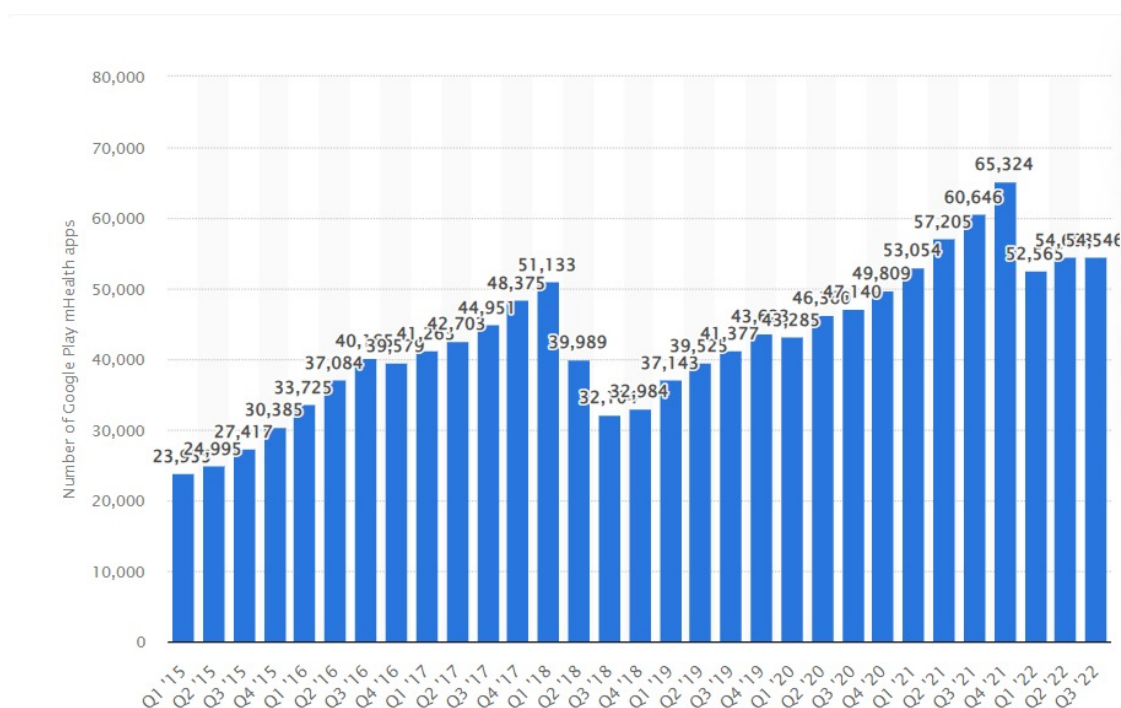


Figura 1.1. Numărul de aplicații mHealth disponibile în Magazinul Google Play din primul trimestru 2015 până în trimestrul 3 2022 [2].

1.2. Tema propusă

Tema lucrării de licență este **Estimarea compoziției corporale utilizând algoritmi de învățare automată**. Obiectivele principale sunt dezvoltarea unui model de învățare automată capabil să estimeze compoziția corporală pe baza unor măsurători antropometrice simple și integrarea acestui model într-o aplicație mobilă Android pentru a permite unui public larg să-l utilizeze.

1.2.1. Obiective și justificarea abordării

Obiectivele specifice includ:

- **Dezvoltarea unui model de regresie liniară** pentru estimarea procentului de grăsime corporală.
- **Integrarea modelului de învățare automată** într-o aplicație mobilă Android.
- **Evaluarea performanței modelului și a aplicației** în condiții reale de utilizare.

Justificarea abordării constă în necesitatea de a oferi utilizatorilor o metodă accesibilă și precisă pentru monitorizarea compoziției corporale, deoarece metodele tradiționale sunt adesea costisitoare și necesită echipamente specializate [1].

Pe baza unor date ușor de colectat, cum ar fi măsurătorile circumferințelor corporale, vârsta, greutatea și înălțimea, algoritmi de învățare automată permit realizarea unor estimări precise. Acest lucru permite realizarea unui echilibru între costuri, accesibilitate și acuratețe, aspecte esențiale pentru adoptarea pe scară largă a unei astfel de tehnologii.

Algoritmi de învățare automată oferă o abordare flexibilă și scalabilă pentru analiza datelor complexe. Aceștia pot învăța și se pot adapta pe baza datelor noi, îmbunătățind continuu acuratețea estimărilor. Apoi, prin utilizarea unui model de regresie liniară, se poate obține o soluție eficientă din punct de vedere computațional, care este potrivită pentru implementarea pe dispozitive mobile cu resurse limitate.

1.3. Realizările actuale pe aceeași temă și analiza tipurilor de produse/ aplicații existente

În prezent, există mai multe aplicații și dispozitive care oferă funcționalități similare, precum balanțele inteligente și diverse aplicații de fitness. De exemplu, **Fitbit Aria** și **MyFitnessCompanion** sunt două produse populare care urmăresc greutatea și procentul de grăsime corporală, dar depind de un hardware suplimentar și pot fi costisitoare [3, 4].

Fitbit Aria [3] (Figura 1.2) este un dispozitiv inteligent de monitorizare a greutății și a procentului de grăsime corporală, lansat de Fitbit. Fitbit, fondată în 2007, a devenit rapid una dintre cele mai importante companii din domeniul dispozitivelor de monitorizare a sănătății și a fitness-ului. Fitbit Aria a fost lansată pentru prima dată în 2012 și reprezintă una dintre primele balanțe inteligente de pe piață, integrându-se perfect în ecosistemul de produse Fitbit.

• Avantaje:

- **Integrare cu ecosistemul Fitbit:** utilizatorii pot sincroniza datele de la Fitbit Aria cu alte dispozitive și aplicații Fitbit, cum ar fi brățările și ceasurile de fitness, pentru a obține o imagine completă a sănătății lor.
- **Colectare precisă a datelor:** Fitbit Aria utilizează senzori avansați de greutate și bioimpedanță pentru a măsura greutatea, procentul de grăsime corporală și indicele de masă corporală (BMI).

• Dezavantaje:

- **Cost ridicat:** prețul dispozitivului poate fi o barieră pentru mulți utilizatori, fiind mai scump față de cântarele tradiționale.
- **Necesită achiziționarea unui dispozitiv separat:** spre deosebire de aplicațiile care utilizează doar smartphone-ul, Fitbit Aria necesită achiziționarea și utilizarea unui cântar inteligent separat.
- **Tehnologii utilizate:**
 - **Senzori de greutate și analiză a bioimpedanței:** utilizați pentru a măsura greutatea și compoziția corporală.
 - **Conectivitate Bluetooth și Wi-Fi:** permite sincronizarea datelor cu aplicația mobilă Fitbit, facilitând accesul la date în timp real.
- **Inovații:**
 - **Integrarea cu alte dispozitive Fitbit:** oferă o soluție completă de monitorizare a sănătății, incluzând urmărirea activității fizice, a somnului și a alimentației, toate centralizate într-o singură platformă.

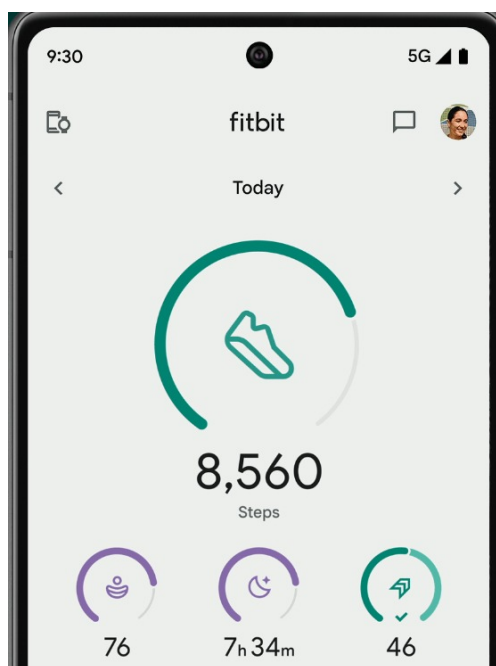


Figura 1.2. Interfața aplicației Fitbit [3].

MyFitnessCompanion [4] este o aplicație mobilă destinată monitorizării sănătății și a activității fizice, dezvoltată pentru a ajuta utilizatorii să-și gestioneze sănătatea și fitness-ul. Aceasta oferă o gamă largă de funcționalități și se integrează cu diverse dispozitive wearable, facilitând colectarea de date din multiple surse.

- **Avantaje:**
 - **Conectivitate cu diverse dispozitive wearable:** aplicația permite utilizatorilor să colecteze date din multiple surse, oferind astfel o imagine cuprinzătoare a sănătății lor.
 - **Colectare cuprinzătoare a datelor:** MyFitnessCompanion monitorizează diverse aspecte ale sănătății, printre care greutatea, activitatea fizică și somnul, oferă statistici detaliate și recomandări personalizate.

- **Dezavantaje:**

- **Necesită multiple dispozitive pentru o funcționalitate completă:** utilizatorii trebuie să dețină mai multe dispozitive pentru a beneficia de toate funcționalitățile oferite de aplicație.

- **Tehnologii utilizate:**

- **API-uri pentru colectarea datelor:** MyFitnessCompanion utilizează diverse API-uri pentru a sincroniza informațiile de la diferite dispozitive wearable; asigură o colectare precisă și centralizată a datelor.
- **Analiza datelor în cloud:** aplicația stochează și analizează datele colectate în cloud, permite accesul în timp real la informații și oferă recomandări personalizate bazate pe date.

- **Inovații:**

- **Integrarea cu multiple platforme:** permite utilizatorilor să centralizeze toate datele de sănătate într-o singură aplicație, facilitând monitorizarea completă a stării de sănătate.

Apple Health [5] este o aplicație dezvoltată de Apple Inc. lansată în 2014 odată cu iOS 8. Aplicația centralizează și organizează datele de sănătate și fitness de pe iPhone și dispozitivele wearable compatibile, oferind utilizatorilor o imagine completă a stării lor de sănătate.

- **Avantaje:**

- **Centralizare a datelor de sănătate:** Apple Health colectează date din diverse aplicații și dispozitive într-o singură platformă, facilitând astfel, accesul rapid la informații complete despre sănătatea utilizatorului.
- **Integrare cu ecosistemul Apple:** aplicația oferă o experiență fluidă utilizatorilor de iPhone și Apple Watch și permite sincronizarea datelor între diferite dispozitive Apple.

- **Dezavantaje:**

- **Limitat la ecosistemul Apple:** utilizatorii de dispozitive non-Apple nu pot beneficia de această aplicație, ceea ce limitează accesibilitatea sa.

- **Tehnologii utilizate:**

- **HealthKit API:** utilizat pentru a colecta și centraliza datele de sănătate și fitness de la diverse aplicații și dispozitive compatibile.
- **Algoritmi de analiză a datelor:** oferă informații personalizate și recomandări utilizatorilor, bazate pe analiza datelor colectate.

Google Fit [6] este o aplicație dezvoltată de Google care a fost lansată în 2014. Aplicația colectează date de sănătate și fitness de la diverse dispozitive și aplicații, oferind utilizatorilor posibilitatea de a-și monitoriza activitatea fizică și starea de sănătate.

- **Avantaje:**

- **Colectare a datelor din diverse surse:** Google Fit permite utilizatorilor să monitorizeze activitatea fizică, somnul și alte aspecte ale sănătății, cu integrarea datelor din multiple surse.

- **Integrare cu diverse dispozitive și aplicații:** aplicația oferă o platformă deschisă pentru monitorizarea sănătății, permite integrarea cu o gamă largă de dispozitive și aplicații.
- **Dezavantaje:**
 - **Funcționalități limitate comparativ cu aplicațiile dedicate de fitness:** Google Fit poate să nu ofere toate funcțiile avansate disponibile în alte aplicații specializate.
- **Tehnologii utilizate:**
 - **Google Fit API:** utilizat pentru a colecta și sincroniza datele de sănătate și fitness de la diverse dispozitive și aplicații.
 - **Algoritmi de analiză a datelor:** oferă utilizatorilor posibilitatea de a-și monitoriza progresul și de a primi sugestii personalizate pentru îmbunătățirea stării de sănătate.
- **Inovații:**
 - **Platformă deschisă:** permite dezvoltatorilor să integreze aplicațiile și dispozitivele lor cu Google Fit, oferind utilizatorilor o experiență unificată și personalizată.

Samsung Health [7] este o aplicație dezvoltată de Samsung și lansată în 2012. Aplicația oferă o gamă largă de funcții pentru monitorizarea sănătății și fitness-ului (Figura 1.3), inclusiv urmărirea activităților fizice, gestionarea greutății, monitorizarea stresului și a somnului.

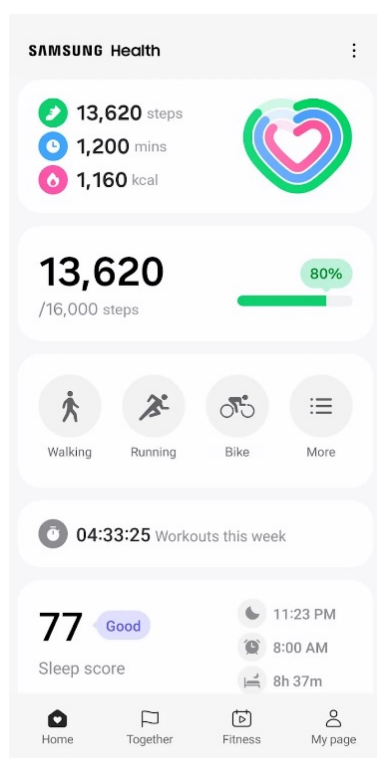


Figura 1.3. Interfața aplicației Samsung Health [7].

- **Avantaje:**
 - **Gama largă de funcții:** Samsung Health monitorizează diverse aspecte ale sănătății, oferind o soluție completă pentru utilizatori.

- **Integrare cu dispozitivele Samsung:** aplicația se sincronizează cu diverse dispozitive Samsung, oferind o experiență holistică și unificată.
- **Dezavantaje:**
 - **Limitată în principal la ecosistemul Samsung:** utilizatorii de dispozitive non-Samsung nu pot beneficia pe deplin de toate funcțiile oferite de Samsung Health.
- **Tehnologii utilizate:**
 - **Senzori de mișcare și sănătate:** utilizați pentru a monitoriza activitatea fizică, ritmul cardiac, somnul și alte aspecte ale sănătății.
 - **Conectivitate Bluetooth și Wi-Fi:** permite sincronizarea datelor cu aplicația mobilă Samsung Health, facilitând accesul în timp real la informații.
- **Inovații:**
 - **Integrare completă cu dispozitivele Samsung:** oferă o experiență unificată, permițând utilizatorilor să monitorizeze diverse aspecte ale sănătății lor într-o singură aplicație.

Strava [8] este o aplicație populară printre sportivi pentru urmărirea activităților de alergare și ciclism. Fondată în 2009, Strava a devenit rapid una dintre cele mai utilizate aplicații pentru monitorizarea performanței sportive.

- **Avantaje:**
 - **Funcții avansate de analiză a performanței:** include segmentarea traseelor, analiză detaliată a performanței și compararea rezultatelor cu alte persoane.
 - **Comunitatea activă de utilizatori:** permite conectarea cu alți sportivi, precum și participarea la competiții virtuale.
- **Dezavantaje:**
 - **Concentrată în principal pe alergare și ciclism:** poate să nu fie la fel de utilă pentru alte tipuri de activități fizice.
- **Tehnologii utilizate:**
 - **GPS și senzori de mișcare:** utilizează GPS-ul pentru a urmări traseele și senzori de mișcare pentru a măsura activitatea fizică.
 - **Algoritmi de analiză a performanței:** oferă feedback detaliat asupra performanței și permite segmentarea traseelor.
- **Inovații:**
 - **Comunitatea activă de utilizatori:** oferă motivație prin competiție și compararea rezultatelor cu alți sportivi.

Fiecare din aceste aplicații demonstrează diversitatea și complexitatea soluțiilor disponibile pe piață, cu avantaje și limitări. Prin analiza acestora, putem identifica bune practici dar și oportunități de îmbunătățire pentru aplicația noastră.

1.3.1. Provocări etice și beneficii ale tehnologiei mHealth

Un studiu publicat în *The American Journal of Medicine* de John P. Higgins explorează utilizarea aplicațiilor mobile pentru sănătatea și fitness-ul pacienților, subliniind potențialul acestora de a îmbunătăți starea de sănătate prin monitorizarea activităților fizice, a dietei și a somnului [1]. Aplicațiile de fitness, cum ar fi cele pentru urmărirea pașilor sau a exercițiilor, utilizează senzori încorporați în smartphone-uri, cum ar fi GPS-ul și accelerometrul, pentru a colecta și analiza datele utilizatorilor.

În cadrul proiectului nostru, aceste tehnologii sunt esențiale pentru monitorizarea precisă și în timp real a activității fizice a utilizatorilor. Integrarea senzorilor de mișcare și a altor tehnologii permite aplicației să ofere o evaluare holistică a stării de sănătate a utilizatorilor, contribuind astfel la atingerea obiectivelor de sănătate și fitness.

Studiul „*Mobile technology identity and self-efficacy: Implications for the adoption*” evidențiază impactul tehnologiei mobile asupra autoeficacității utilizatorilor și adoptării acesteia în diverse domenii, inclusiv sănătatea digitală [9]. Cercetările arată că utilizatorii care se simt confortabili cu tehnologia mobilă sunt mai predispuși să adopte și să utilizeze în mod eficient aplicații de sănătate și fitness.

Această descoperire subliniază importanța designului intuitiv și accesibil al aplicației noastre, care trebuie să fie ușor de utilizat pentru a maximiza adoptarea și utilizarea de către un public larg. Prin asigurarea unei experiențe prietenoase, aplicația poate contribui la creșterea autoeficacității, precum și la îmbunătățirea stării de sănătate a utilizatorilor.

Articolul „*Exploring benefits and ethical challenges in the rise of mHealth (mobile healthcare) technology*” examinează avantajele și provocările etice ale tehnologiei mHealth [10]. Beneficiile includ accesibilitatea sporită la informații de sănătate și monitorizarea continuă a stării de sănătate, în timp ce provocările etice se referă la confidențialitatea datelor și la securitatea informațiilor personale ale utilizatorilor.

Pentru a aborda aceste provocări, aplicațiile trebuie să utilizeze servicii de securitate și criptare pentru a proteja datele utilizatorilor. În plus, politica de confidențialitate a aplicațiilor trebuie concepută astfel încât să ne asigurăm că utilizatorii sunt informați cu privire la modul în care sunt gestionate datele lor și că au control asupra informațiilor personale pe care le partajează.

Analiza conținutului aplicațiilor de sănătate și fitness dezvăluie că majoritatea aplicațiilor includ funcționalități de monitorizare a activității fizice, stabilirea de obiective și feedback personalizat. Aplicațiile mai avansate oferă și funcționalități de coaching virtual, sincronizare cu alte dispozitive și integrare cu rețele sociale pentru a încuraja utilizatorii să-și atingă obiectivele de sănătate și fitness.

1.4. Elaborarea specificațiilor privind caracteristicile așteptate de la aplicație

În această secțiune, vom detalia specificațiile și caracteristicile așteptate de la aplicația dezvoltată pentru estimarea compoziției corporale utilizând algoritmi de învățare automată.

1.4.1. Tehnologii folosite pentru implementare

Java pentru dezvoltarea aplicației Android [11]: Java este limbajul de programare principal utilizat pentru dezvoltarea aplicației mobile. Acesta oferă stabilitate și compatibilitate excelentă cu platforma Android, permițând implementarea eficientă a funcționalităților aplicației.

Firestore pentru autentificare și stocare: Firestore [12] este o platformă de dezvoltare a aplicațiilor mobile și web care oferă o gamă largă de servicii ce facilitează implementarea rapidă și securizată a funcționalităților esențiale.

Firestore Authentication: permite autentificarea utilizatorilor prin diverse metode (email și parolă, Google Sign-In).

Firestore Realtime Database: o bază de date NoSQL care stochează datele în format

JSON și sincronizează modificările în timp real.

Firestore Storage: utilizat pentru stocarea fișierelor mari, cum ar fi imagini și videoclipuri, asigurând securitatea și accesibilitatea acestora.

Scikit-learn și Pandas pentru modelul de învățare automată: Scikit-learn și Pandas [13] sunt biblioteci Python esențiale pentru dezvoltarea și evaluarea modelului de regresie liniară utilizat pentru estimarea procentului de grăsime corporală. Scikit-learn oferă instrumente pentru preprocesarea datelor, antrenarea modelului și evaluarea performanței acestuia. Pandas facilitează manipularea și analiza seturilor de date, permițând pregătirea eficientă a datelor pentru modelul de învățare automată.

API-uri: integrarea cu API-uri pentru articole relevante sau pentru localizare și cartografiere precum cele oferite de Google Maps API, ce permit aplicațiilor să afișeze hărți.

1.4.2. Caracteristicile așteptate de la aplicație

Aplicația își propune atingerea următoarelor aspecte:

- **Autentificarea și gestionarea profilului utilizatorului:** sunt esențiale pentru a oferi o experiență personalizată și securizată. Utilizatorii pot crea conturi noi, se pot autentifica folosind diverse metode și își pot gestiona informațiile personale.
- **Introducerea și stocarea datelor antropometrice:** permite utilizatorilor să introducă și să stocheze măsurători personale necesare pentru calculul compoziției corporale, cum ar fi greutatea și măsurătorile corpului.
- **Estimarea procentului de grăsime corporală utilizând modelul de învățare automată:** reprezintă una dintre funcționalitățile centrale ale aplicației: estimarea procentului de grăsime corporală se realizează pe baza măsurătorilor antropometrice furnizate de utilizatori.
- **Monitorizarea pașilor zilnici și calculul necesarului caloric zilnic:** utilizând senzorii dispozitivului mobil (accelerometru, giroscop), aplicația contorizează pașii zilnici efectuați de utilizatori. Datele sunt stocate și sincronizate în Firebase Realtime Database, asigurând accesul facil la istoricul activităților. Totodată, aplicația va calcula necesarul caloric zilnic pe baza obiectivelor setate și a nivelului de activitate fizică.
- **Integrarea cu alte servicii**
- **Calcularea BMI-ului (Body Mass Index)**

Aceste funcționalități oferă utilizatorilor informații valoroase, personalizate pentru gestionarea sănătății și condiției fizice.

Aplicația își propune totodată atingerea următoarele aspecte:

- **Accesibilitate și ușurință în utilizare**

Aplicația va fi concepută pentru a fi accesibilă, intuitivă, oferind utilizatorilor o experiență plăcută și ușor de navigat. Interfața de utilizare (UI) este proiectată pentru a facilita introducerea datelor și accesarea funcționalităților principale fără eforturi suplimentare din partea utilizatorului. Astfel, vom avea meniuri clare, formulare ușor de completat, grafice ale datelor de sănătate și fitness, facilitând înțelegerea progresului și motivarea utilizatorilor pentru atingerea obiectivelor.

- **Precizie și securitate**

Modelul de învățare automată trebuie să ofere estimări precise ale compoziției corporale, bazate pe datele introduse de utilizatori. De asemenea, aplicația trebuie să asigure securitatea datelor utilizatorilor, protejând informațiile personale împotriva accesului neautorizat.

Datele sunt criptate atât în tranzit, cât și în repaus, iar accesul la acestea este strict controlat. Utilizatorii vor putea gestiona permisiunile, precum și accesul la datele lor personale, având posibilitatea de a vizualiza, actualiza sau șterge informațiile stocate.

- **Scalabilitate și interoperabilitate**

Aplicația trebuie să fie scalabilă astfel încât să poată gestiona un număr mare de utilizatori și date fără a compromite performanța. În plus, trebuie să fie interoperabilă, permițând integrarea cu alte aplicații și dispozitive de sănătate pentru a permite o monitorizare completă a stării de sănătate a utilizatorilor.

1.5. Tendințe și inovații în aplicațiile de sănătate și fitness

Pe măsură ce tehnologia avansează, aplicațiile de sănătate și fitness devin din ce în ce mai sofisticate, integrând tehnologii emergente precum inteligența artificială, învățarea automată și Internetul lucrurilor (IoT). Aceste inovații oferă noi oportunități pentru monitorizarea sănătății și personalizarea experienței utilizatorilor.

1.5.1. Inteligența artificială și învățarea automată

Aplicațiile moderne de sănătate și fitness utilizează algoritmi de inteligență artificială și învățare automată pentru a analiza datele utilizatorilor și a oferi recomandări personalizate. Aceste tehnologii permit detectarea modelelor și tendințelor în comportamentul utilizatorilor, contribuind la îmbunătățirea stării de sănătate prin intervenții proactive și feedback personalizat [14].

1.5.2. Internetul lucrurilor (IoT)

Integrarea dispozitivelor IoT în aplicațiile de sănătate și fitness permite colectarea continuă a datelor de la diverse surse, cum ar fi brățările de fitness, ceasurile inteligente și senzorii de sănătate. Aceste date sunt analizate în timp real pentru a oferi o imagine completă și actualizată a stării de sănătate a utilizatorilor [15].

1.5.3. Realitatea augmentată și virtuală

Aplicațiile de sănătate și fitness încep să exploreze utilizarea realității augmentate (AR) și virtuale (VR) pentru a crea experiențe interactive și captivante. Aceste tehnologii pot fi utilizate pentru antrenamente ghidate, simulări de exerciții și monitorizarea posturii, oferind utilizatorilor o modalitate inovatoare de a-și atinge obiectivele de fitness [16, 17].

1.5.4. Monitorizarea sănătății în timp real

Aplicațiile de sănătate și fitness au început să folosească monitorizarea sănătății în timp real pentru a oferi informații actualizate despre starea utilizatorilor. Aceasta include monitorizarea continuă a nivelului de oxigen în sânge, temperaturii corpului, ritmului cardiac dar și a altor parametri fiziologici importanți. Datele colectate sunt utilizate pentru a oferi feedback imediat, potențiale probleme de sănătate putând fi detectate înainte ca să devină critice [18].

1.5.5. Personalizarea experienței utilizatorilor

Un aspect cheie al aplicațiilor moderne de sănătate și fitness este capacitatea de a personaliza experiența utilizatorilor. Prin analiza datelor colectate și utilizând algoritmi de învățare automată, aplicațiile pot oferi recomandări personalizate pentru dietă, exerciții fizice și alte activități de sănătate. Acest nivel de personalizare crește angajamentul utilizatorilor și îmbunătățește eficacitatea intervențiilor de sănătate [19].

1.6. Concluzii

Dezvoltarea unei aplicații mobile pentru estimarea compoziției corporale care utilizează algoritmi de învățare automată, reprezintă o abordare inovatoare și accesibilă pentru monitorizarea sănătății. Aplicația propusă oferă o soluție scalabilă și personalizată pentru utilizatori prin utilizarea tehnologiilor moderne și a serviciilor cloud. Analiza aplicațiilor existente și a tendințelor emergente subliniază cât de important este designul centrat pe utilizator și inovația atunci când vine vorba de produse de sănătate și fitness.

Prin abordarea propusă, ne propunem să contribuim la îmbunătățirea sănătății populației prin furnizarea unui instrument ușor de utilizat, precis și accesibil pentru monitorizarea compoziției corporale și a altor parametri de sănătate. Aplicația noastră oferă, de asemenea, un exemplu de bune practici în dezvoltarea de aplicații mobile de sănătate care combină abordările etice necesare și tehnologiile emergente pentru a garanta succesul pe termen lung în domeniu.

Capitolul 2. Proiectarea aplicației

2.1. Concepte generale

Aplicația pentru estimarea grăsimii corporale utilizând algoritmi de învățare automată, de monitorizare a necesarului caloric, a pașilor și a indicelui de masă corporală este proiectată pentru a oferi utilizatorilor un instrument comprehensiv de monitorizare a sănătății și fitnessului. Aceasta integrează funcționalități multiple care permit utilizatorilor să își urmărească progresul în timp real, să acceseze informații relevante și să primească recomandări personalizate.

2.1.1. Obiectivele aplicației

Aplicația are ca obiective principale furnizarea unor instrumente accesibile, precise pentru monitorizarea sănătății. Printre acestea se numără:

Estimarea compoziției corporale: utilizarea algoritmilor de învățare automată pentru a estima grăsimea corporală și alte metrici importante pe baza măsurătorilor corporale furnizate de utilizatori.

Monitorizarea activităților fizice: urmărirea numărului de pași parcurși zilnic și oferirea de grafice de progres și statistici detaliate.

Accesarea informațiilor relevante: afișarea articolelor de știri relevante pentru utilizatori, precum și a locațiilor sălilor de sport din apropiere.

Autentificarea utilizatorilor: gestionarea securizată a autentificării utilizatorilor folosind Firebase Authentication.

Personalizarea experienței utilizatorilor: stocarea și sincronizarea datelor utilizatorilor în Firebase Realtime Database pentru a permite accesul și actualizarea acestora de pe diferite dispozitive.

2.2. Arhitectura aplicației

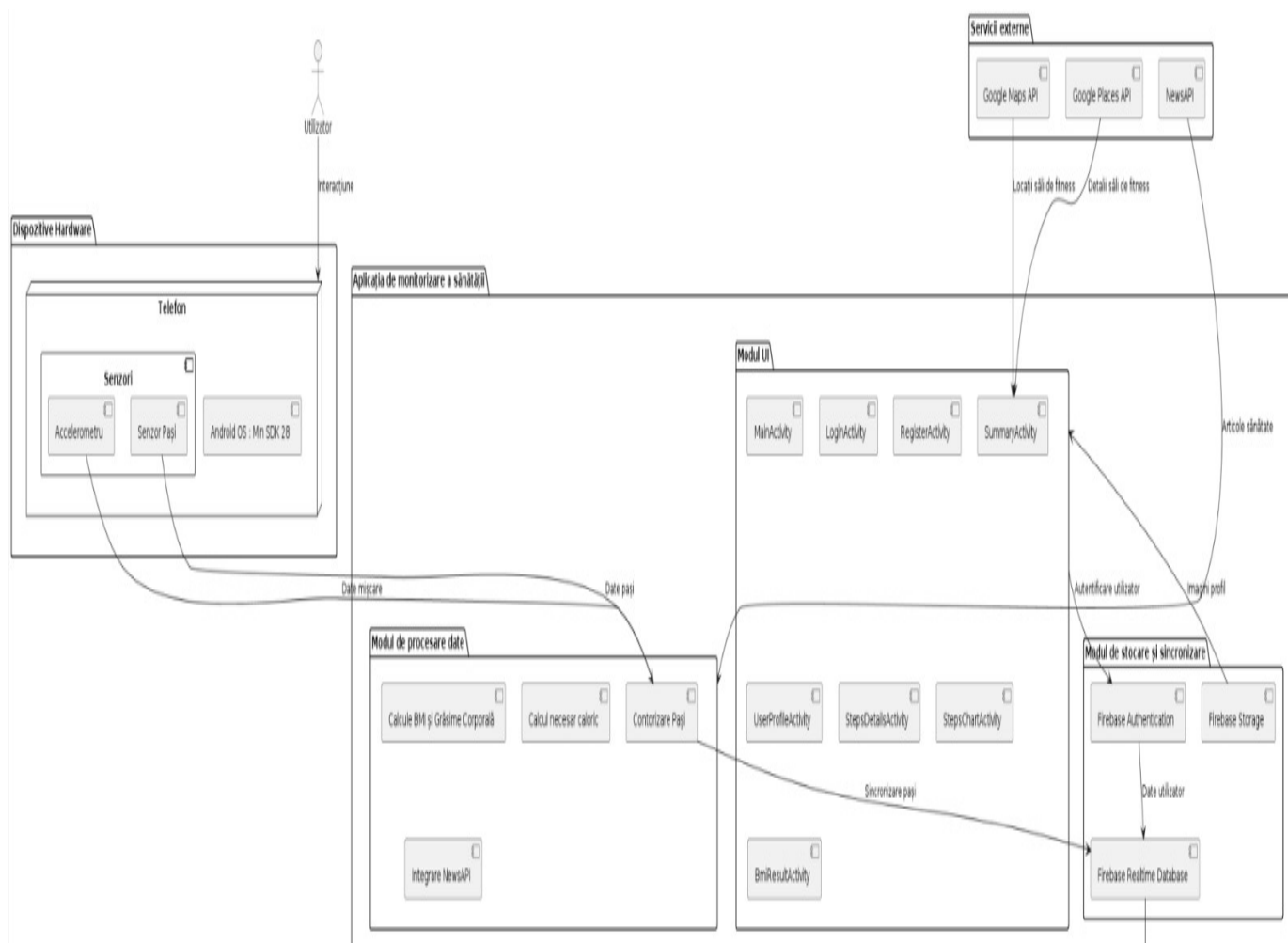


Figura 2.1. Arhitectura aplicației

2.2.1. Analiza platformei hardware

Aplicația va rula pe dispozitive mobile cu sistem de operare Android. Aceste dispozitive trebuie să aibă acces la internet pentru a putea comunica cu serviciile Firebase pentru autentificare și stocare a datelor. În plus, senzorii dispozitivului, cum ar fi accelerometrul și giroscopul, sunt utilizați pentru a colecta date despre activitatea fizică a utilizatorilor.

2.2.2. Analiza platformei software

Aplicația este dezvoltată folosind Android Studio, iar limbajul principal de programare este Java. Pentru gestionarea autentificării și stocării datelor, aplicația utilizează Firebase Authentication și Firebase Realtime Database. Datele colectate de la utilizatori sunt salvate și prelucrate pentru a oferi estimări precise ale indicatorilor de sănătate, cum ar fi BMI și procentul de grăsime corporală. Platforma Android oferă o varietate de caracteristici și API-uri care facilitează dezvoltarea aplicațiilor de sănătate și fitness.

2.2.3. Avantaje

- **Accesibilitate largă:** Android este disponibil pe o gamă largă de dispozitive de la diferiți producători.
- **API-uri extinse:** Android oferă API-uri pentru senzori, gestionarea activităților și alte

funcționalități esențiale pentru aplicațiile de sănătate [11].

- **Compatibilitate:** platforma Android permite dezvoltatorilor să creeze aplicații care rulează pe o varietate de dispozitive cu specificații hardware diferite.
- **Flexibilitate în dezvoltare:** Android Studio oferă un mediu de dezvoltare integrat (IDE) puternic și flexibil care permite dezvoltatorilor să creeze aplicații complexe cu ușurință.
- **Acces la hardware:** Android permite accesul direct la hardware-ul dispozitivului, cum ar fi senzori de mișcare, GPS și camere, care sunt esențiali pentru astfel de programe.
- **Suport comunitar:** Android are o comunitate vastă de dezvoltatori care contribuie cu biblioteci, tutoriale și suport tehnic.
- **Scalabilitate:** Firebase oferă servicii scalabile pentru autentificare și stocare de date [12].
- **Integrare ușoară:** Firebase oferă o integrare simplă și rapidă cu aplicațiile Android, facilitând implementarea funcționalităților complexe precum autentificarea sau stocarea datelor.

2.2.4. Limitări și dezavantaje

- **Fragmentarea dispozitivelor**

Android este cunoscut pentru fragmentarea sa, existând o mare diversitate de dispozitive cu specificații hardware și versiuni de sistem de operare variate. Acest lucru poate duce la probleme de compatibilitate și la necesitatea unor eforturi suplimentare pentru testarea și optimizarea aplicației pe diferite dispozitive. Fragmentarea poate afecta și performanța aplicației, în special pe dispozitivele mai vechi sau mai puțin performante.

- **Accesul la hardware și permisiuni**

Accesul la anumite funcționalități hardware, cum ar fi senzorii de mișcare și modulul GPS, necesită permisiuni speciale din partea utilizatorilor. Gestionarea acestor permisiuni poate fi complexă, iar utilizatorii pot fi reticenți în a acorda permisiunile necesare, ceea ce poate limita funcționalitatea aplicației. Calitatea datelor colectate poate fi afectată deoarece acuratețea și performanța senzorilor poate varia de la un dispozitiv la altul.

- **Dependența de platformă:**

Aplicația este limitată la dispozitivele Android, ceea ce poate exclude utilizatorii altor platforme mobile.

2.3. Achiziția și salvarea datelor

2.3.1. Descriere generală

Modulul de achiziție a datelor din aplicație are rolul de a colecta și gestiona datele referitoare la compoziția corporală a utilizatorilor. Acesta utilizează un set de algoritmi de învățare automată pentru a estima grăsimea corporală pe baza măsurătorilor corporale colectate de la utilizatori. Datele sunt procesate și utilizate pentru a oferi predicții precise.

Datele sunt colectate de la utilizatori prin intermediul unei interfețe de utilizare intuitive. Utilizatorii introduc măsurători ale circumferințelor corporale și alte date relevante, care sunt apoi folosite pentru a estima compoziția corporală. Datele sunt stocate în Firebase Firestore pentru a asigura accesibilitatea și sincronizarea între diferitele dispozitive ale utilizatorului.

2.3.2. Setul de date utilizat

Setul de date utilizat pentru antrenarea algoritmilor de învățare automată conține estimări ale procentului de grăsime corporală determinate prin cântărire subacvatică și diverse măsurători ale circumferințelor corporale pentru 252 de bărbați [20]. Aceste date au fost colectate pentru a ilustra tehnicile de regresie multiplă și pentru a oferi metode ușoare și necostisitoare de estimare a grăsimii corporale care să nu implice metodele tradiționale mai costisitoare și incomode.

Standardele de măsurare sunt aparent cele enumerate în Benhke și Wilmore (1974), pp. 45-48 [21] unde, de exemplu, circumferința abdomenului este măsurată „lateral, la nivelul creștelor iliace, și anterior, la ombilic”.

Aceste date sunt utilizate pentru a produce ecuațiile predictive pentru greutatea corporală slabă, prezentate în rezumatul „Ecuația de predicție generalizată a compoziției corporale pentru bărbați folosind tehnici simple de măsurare”, K.W. Penrose, A.G. Nelson, A.G. Fisher, FACSM, Centrul de Cercetare a Performanței Umane, Universitatea Brigham Young, Provo, Utah 84602, așa cum este listat în Medicine and Science in Sports and Exercise, vol. 17, nr. 2, aprilie 1985, p. 189 [22] (Ecuația predictivă a fost obținută din primele 143 din cele 252 de cazuri enumerate mai jos).

2.3.3. Variabilele setului de date

Setul de date include următoarele variabile:

- **Density:** densitatea corporală determinată prin cântărire subacvatică.
- **BodyFat:** procentul de grăsime corporală calculat folosind ecuația lui Siri (1956) Ec.(2.1)[23].

$$\%BF = \frac{495}{D} - 450 \quad (2.1)$$

- **Age:** vârsta subiectului (ani).
- **Weight:** greutatea subiectului (lbs).
- **Height:** înălțimea subiectului (inch).
- **Neck, Chest, Abdomen, Hip, Thigh, Knee, Ankle, Biceps, Forearm, Wrist:** Circumferințele diferitelor părți ale corpului (cm)

2.3.4. Prelucrarea și analiza datelor

Prelucrarea datelor implică eliminarea outlier-ilor și normalizarea datelor pentru a asigura acuratețea modelului de învățare automată.

2.3.5. Integrarea cu Firebase

Firebase Realtime Database este utilizat pentru a stoca datele utilizatorilor, asigurând sincronizarea și accesibilitatea acestora pe diferite dispozitive. Realtime Database permite o gestionare eficientă a datelor și suportă operații de interogare complexe, ceea ce facilitează recuperarea rapidă a datelor necesare pentru predicții și analiză. Utilizatorii își pot accesa datele în timp real, iar orice modificare este imediat reflectată pe toate dispozitivele conectate.

2.3.6. Monitorizarea activităților fizice

Clasa StepCounterService utilizează senzorii dispozitivului pentru a monitoriza numărul de pași parcurși de utilizator. Datele colectate sunt stocate în Firebase Realtime Database și sunt utilizate pentru a oferi statistici detaliate și grafice de progres. Aceasta permite utilizatorilor să-și urmărească activitatea fizică zilnică și să își stabilească obiective de fitness.

2.3.7. API-ul pentru știri

Clasa ArticleApiService interacționează cu un API extern pentru a obține articole de știri relevante pentru utilizatori. Articolele sunt afișate în interfața utilizatorului prin intermediul ArticleAdapter și sunt gestionate în ArticleRepository.

2.3.8. API-ul Google pentru locație și harta sălilor

Clasa FetchNearbyGymsTask utilizează API-ul Google Maps pentru a obține locațiile sălilor de sport din apropierea utilizatorilor. Aceste locații sunt afișate pe o hartă interactivă în aplicație, permițând utilizatorilor să găsească rapid și ușor săli de sport în apropierea lor.

2.4. Componenta software

2.4.1. Arhitectura software a aplicației

Aplicația este construită folosind Java pentru dezvoltarea Android și Python pentru implementarea algoritmilor de învățare automată. Firebase este utilizat pentru autentificarea utilizatorilor și stocarea datelor. Aplicația adoptă arhitectura Model-View-ViewModel (MVVM) pentru partea centrală a aplicației și anume clasa SummaryActivity în cadrul căreia sunt gestionate fragmentele pentru principalele funcționalități. MVVM separă clar logica de prezentare de logica de afaceri și de accesul la date. Beneficiile modelului MVVM includ componente reutilizabile, dezvoltarea independentă a interfeței grafice și a logicii de business sau back-end, flexibilitatea de a modifica interfața grafică fără a fi nevoie de schimbarea logicii de business sau back-end, ușurința testării unitare și întreținerea mai ușoară a aplicației [24].

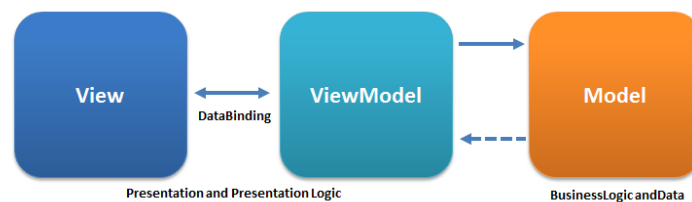


Figura 2.2. Arhitectura Model-view-viewmodel (MVVM)" [25].

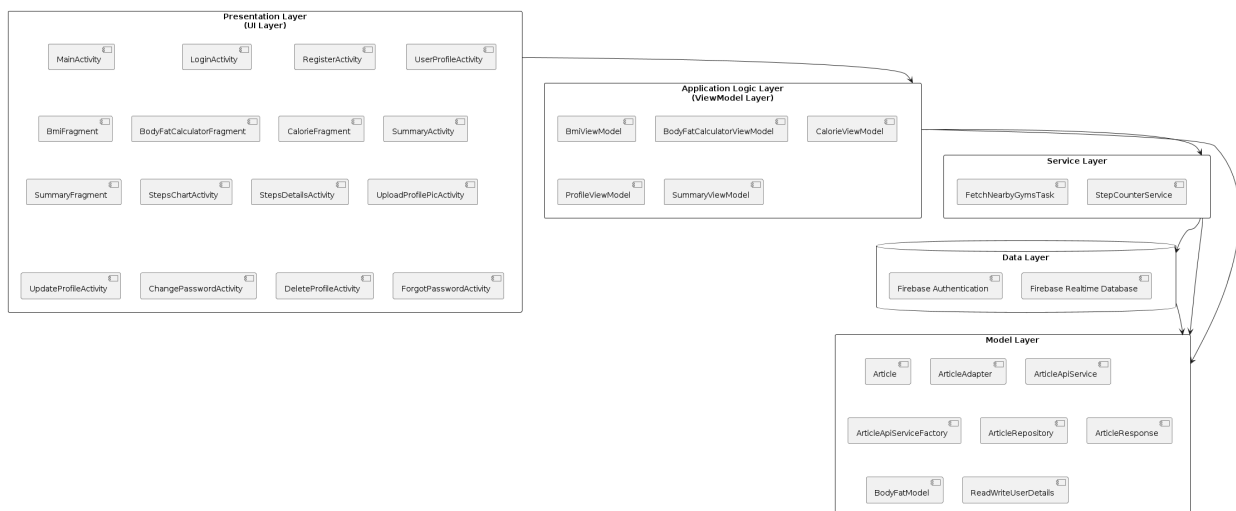


Figura 2.3. Arhitectura software a aplicației.

2.4.2. Diagrame UML

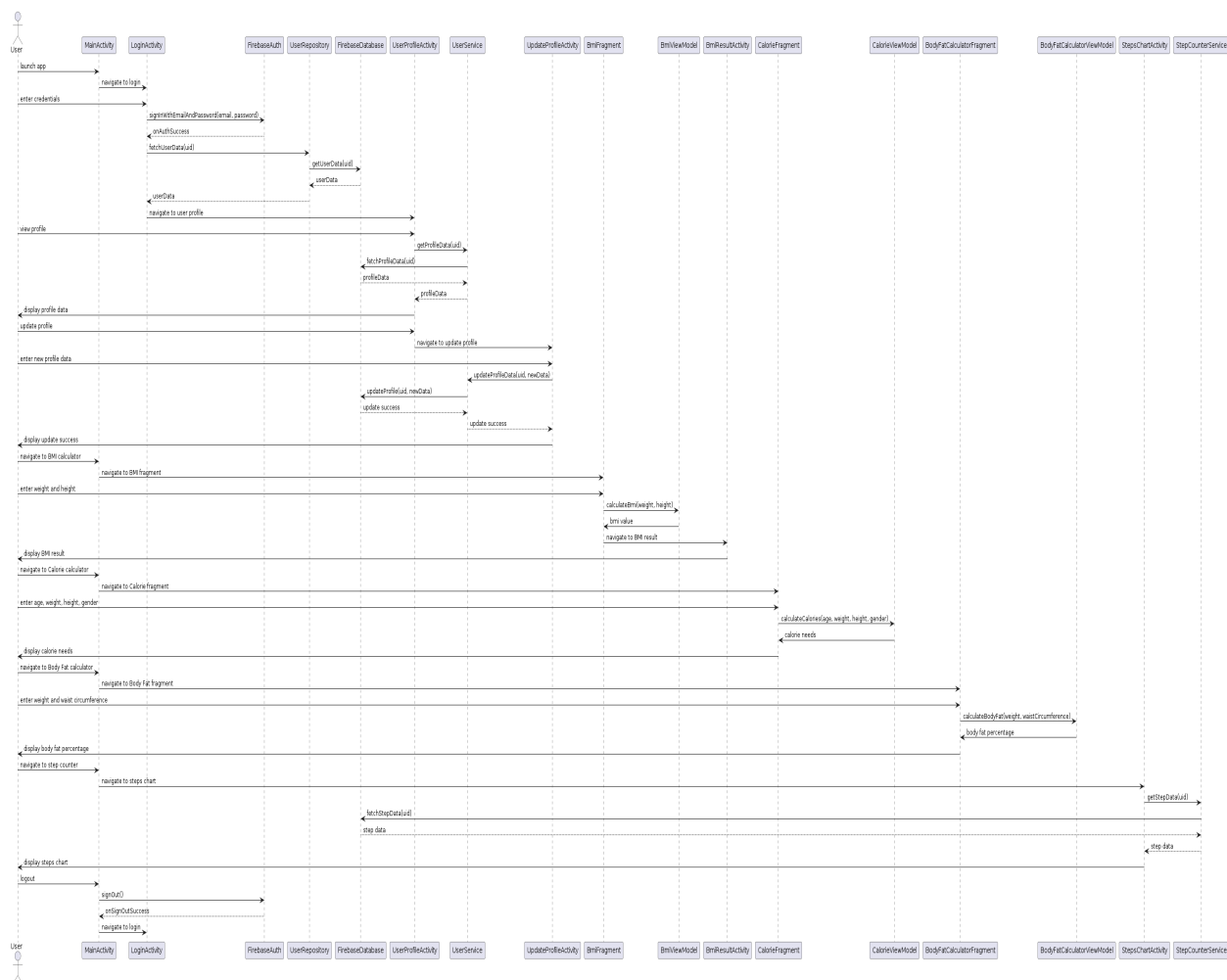


Figura 2.4. Diagrama de secvență completă pentru funcționalitățile aplicației

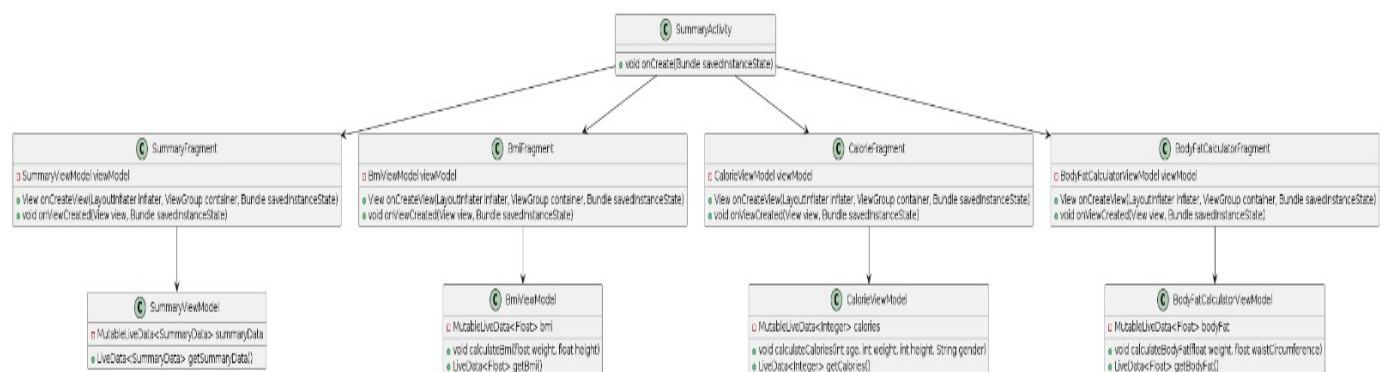


Figura 2.5. Diagrama de clase asociată principalelor funcționalități

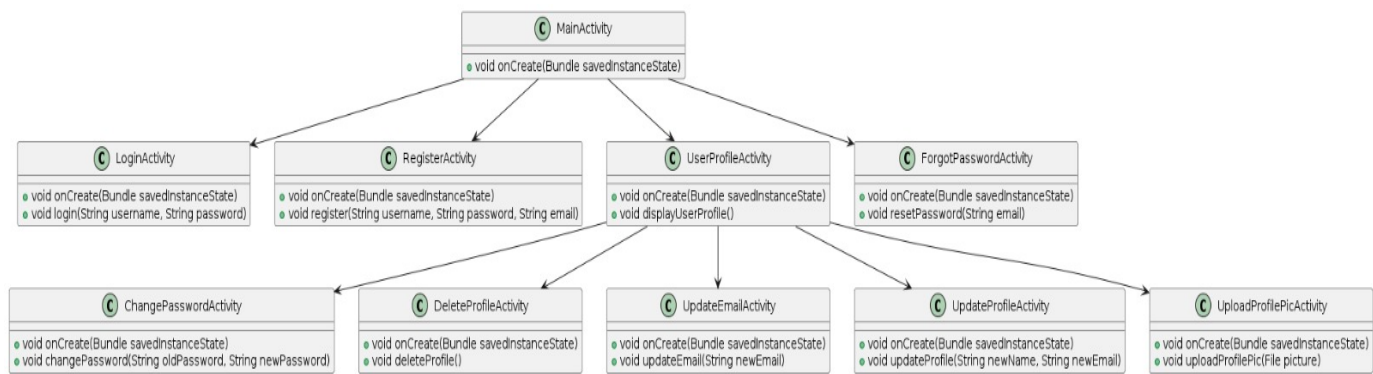


Figura 2.6. Diagrama de clase pentru activitățile asociate contului de utilizator

- **Modelul (Model)**

Modelul conține datele aplicației și logica de manipulare a datelor.

- **View-ul (View)**

View-ul reprezintă interfața utilizator și este responsabil de afișarea datelor și interacțiunile utilizatorului.

- **ViewModel-ul (ViewModel)**

ViewModel-ul acționează ca un intermediar între Model și View, gestionând logica de prezentare a datelor. Acesta include clasele care conțin logica de calcul și manipulare a datelor pentru interfețele utilizator.

- **Fluxul de date**

Fluxul de date în aplicație urmează pașii:

Utilizatorul introduce datele antropometrice în interfața aplicației (View). Datele sunt transmise către ViewModel, unde sunt procesate și validate. ViewModel-ul apelează metodele din Model pentru a calcula procentul de grăsime corporală. Rezultatele calculului sunt transmise înapoi către View pentru a fi afișate utilizatorului. Această separare a preocupărilor permite o dezvoltare mai ușoară și mentenanță a codului, reducând complexitatea și posibilitatea de erori.

2.4.3. Modulele generale ale aplicației și interacțiunile dintre ele

Aplicația este organizată în mai multe module, fiecare având responsabilități distincte. Modulele principale ale aplicației includ:

- **Modulul de autentificare:** gestionarea autentificării utilizatorilor folosind Firebase Authentication [12]. Clasa LoginActivity gestionează autentificarea utilizatorilor. Clasa RegisterActivity gestionează înregistrarea noilor utilizatori. Clasa ForgotPasswordActivity oferă funcționalitatea de resetare a parolei.
- **Modulul de profil utilizator:** gestionarea detaliilor și setărilor profilului utilizatorului. Clasa UserProfileActivity este responsabilă pentru afișarea și actualizarea profilului utilizatorului. ChangePasswordActivity gestionează schimbarea parolei. UpdateEmailActivity permite utilizatorilor să își actualizeze adresa de email. UpdateProfileActivity se ocupă de schimbările aduse datelor utilizatorului.

- **Modulul de monitorizare a sănătății:** colectarea și stocarea datelor despre sănătatea utilizatorilor, inclusiv compoziția corporală, BMI-ul, numărul de pași și necesarul caloric. SummaryFragment este clasa care gestionează interfața de utilizator pentru afișarea sumarelor. SummaryViewModel conține datele pe care fragmentul le afișează și logica de procesare necesară. Totodată, StepCounterService este un serviciu care rulează în fundal și monitorizează pașii utilizatorului.
- **Modulul de estimare a compoziției corporale:** utilizarea unui model de învățare automată pentru estimarea procentului de grăsime corporală [13]. BodyFatCalculatorFragment este clasa responsabilă pentru interfața de utilizator pentru calcularea procentajului de grăsime corporală. BodyFatCalculatorViewModel include metodele de calcul al grăsimii corporale bazate pe greutate și circumferința taliei.
- **Modulul de vizualizare a datelor:** prezentarea grafică a datelor colectate și a rezultatelor estimărilor utilizând grafică interactivă. Este reprezentat de clasele StepsDetailsActivity și StepsChartActivity care se ocupă de vizualizarea prin grafice a datelor despre pași înregistrate în zilele trecute.
- **Modulul de calcul a BMI-ului (Body Mass Index):** utilizarea unei formule pentru calculul acestui index și oferirea unui feedback pe baza rezultatului. BmiFragment gestionează interfața pentru calcularea și afișarea BMI-ului. BmiViewModel include metodele de calcul al BMI-ului bazate pe greutate și înălțime.

Ecuția 2.2 reprezintă modalitatea de calcul a acestui indice.

$$BMI = \frac{weight \ (kg)}{height^2 \ (m^2)} \quad (2.2)$$

- **Modulul de calcul a necesarului caloric:** utilizarea unei formule pentru calculul necesarului caloric zilnic ajustat în funcție de obiectivele și gradul de activitate fizică a utilizatorului. În funcție de scopul utilizatorului (slăbit/îngrășat) se vor genera știri relevante pentru acesta. Clasa CalorieFragment este responsabil pentru interfața de utilizator care afișează informații despre kaloriile consumate. CalorieViewModel conține logica de calcul pentru necesarul caloric zilnic în funcție de vârstă, greutate, înălțime și sex.

2.4.4. Gestionarea autentificării utilizatorilor/Interacțiunea cu Firebase

2.4.4.1. Descriere generală

Modulul de autentificare permite utilizatorilor să se înregistreze și să se autentifice în aplicație utilizând adresa de email și parola. Acest modul este construit folosind Firebase Authentication, care oferă un sistem robust și securizat pentru gestionarea conturilor utilizatorilor. Utilizatorii au posibilitatea de a crea un cont nou, de a se autentifica în contul existent, de a reseta parola în caz de uitare și de a-și actualiza datele personale.

Firebase Authentication asigură o gestionare eficientă a autentificării și oferă funcționalități avansate precum verificarea email-ului și autentificarea prin diferite metode (email, Google, Facebook etc.), însă în această aplicație se utilizează doar autentificarea prin email și parolă.

2.4.4.2. Înregistrarea utilizatorilor

Pentru a se înregistra, utilizatorii trebuie să furnizeze o adresă de email validă și o parolă. Procesul de înregistrare este gestionat de Firebase Authentication, care creează un cont pentru utilizator și stochează informațiile de autentificare în mod securizat.

2.4.4.3. Autentificarea utilizatorilor

Utilizatorii se pot autentifica în aplicație folosind adresa de email și parola furnizate în timpul înregistrării. Firebase Authentication verifică acreditările și permite accesul utilizatorilor la aplicație dacă acestea sunt valide.

2.4.4.4. Resetarea parolei

În cazul în care utilizatorii uită parola, aceștia pot solicita resetarea acesteia prin intermediul aplicației. Firebase Authentication trimite un email de resetare a parolei la adresa de email furnizată de utilizator.

2.4.4.5. Actualizarea datelor utilizatorului

Utilizatorii își pot actualiza informațiile personale, cum ar fi adresa de email sau parola, genul, data de naștere, numele, imaginea de profil direct din aplicație. Firebase Authentication oferă metode pentru actualizarea securizată a acestor date.

2.4.4.6. Gestionarea datelor în Firebase Realtime Database

Datele utilizatorilor autentificați sunt stocate în Firebase Realtime Database. Acest lucru permite sincronizarea și accesibilitatea datelor pe diferite dispozitive. De exemplu, datele de profil ale utilizatorului, numărul de pași, nevoile calorice și alte informații relevante sunt stocate și accesibile în timp real.

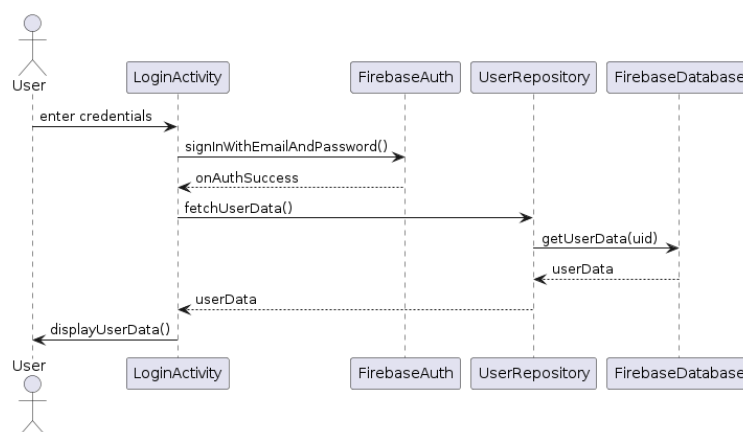


Figura 2.7. Diagrama de secvență pentru autentificarea utilizatorului și obținerea datelor din Firebase

2.4.5. Limitările utilizării serviciilor Firebase

- **Dependența de conexiunea la internet**

Firebase Realtime Database și Firebase Authentication necesită o conexiune stabilă la internet pentru a funcționa corect. În absența unei conexiuni la internet, utilizatorii nu pot accesa sau sincroniza datele, ceea ce poate limita utilizarea aplicației în zone cu conectivitate redusă. Aceasta poate afecta experiența utilizatorilor și utilitatea aplicației în astfel de condiții.

- **Limitările privind securitatea și confidențialitatea datelor**

Deși Firebase oferă funcționalități avansate de securitate, cum ar fi autentificarea utilizatorilor și stocarea criptată a datelor, există riscuri asociate cu stocarea datelor personale în cloud. În caz de breșe de securitate, datele utilizatorilor pot fi compromise. Este esențial să implementăm măsuri suplimentare de securitate.

2.5. Modulul de estimare a procentajului de grăsime

Proiectul se concentrează pe utilizarea unui model de regresie liniară pentru a prezice procentul de grăsime corporală (Body Fat Percentage - BFP) pe baza a două măsurători antropometrice: greutatea și circumferința abdomenului [26]. Alegerea acestui model simplu, dar eficient, este bazată pe capacitatea sa de a oferi predicții precise și rapide, esențiale pentru implementarea într-o aplicație mobilă Android.

2.5.1. Avantajele Metodei OLS și Regresiei Liniare

- **Simplitate și interpretabilitate:**

Regresia liniară este una dintre cele mai simple și mai interpretabile metode de învățare automată. Coeficienții regresiei liniare pot fi interpretați direct, oferind o înțelegere clară a influenței fiecărui predictor asupra variabilei țintă (BFP). Formula pentru regresia OLS (Ordinary Least Squares) este dată de ecuația (2.3)

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \dots + \beta_n x_n + \varepsilon \quad (2.3)$$

unde y este variabila dependentă, x_1, x_2, \dots, x_n sunt variabilele independente, $\beta_0, \beta_1, \dots, \beta_n$ sunt coeficienții estimați, iar ε este eroarea de regresie.

- **Eficiență computațională:**

Modelele de regresie liniară sunt rapide de antrenat și de evaluat, ceea ce le face potrivite pentru aplicațiile mobile unde resursele de calcul sunt limitate.

- **Performanță bună cu date puține:**

Regresia liniară poate oferi performanțe bune chiar și pe seturi de date relativ mici, cum este cazul setului de date utilizat în acest proiect (aproximativ 250 de intrări).

2.5.2. Procesul de Curățare a Datelor

2.5.2.1. Identificarea și eliminarea outlier-ilor

Am identificat outlier-ii pe baza diferenței dintre grăsimea corporală prezisă și cea măsurată și i-am eliminat pentru a îmbunătăți calitatea modelului de predicție.

2.5.2.2. Standardizarea datelor

Am aplicat standardizarea pentru a asigura că toți predictorii au aceeași scară, ceea ce este crucial pentru stabilitatea și performanța modelului de regresie liniară.

2.5.3. Evaluarea modelului

Modelul a fost evaluat folosind metrice precum Mean Absolute Error (MAE), Mean Squared Error (MSE) și coeficientul de determinare (R^2). Aceste metrice oferă o imagine clară a performanței modelului, unde un R^2 ridicat indică o bună potrivire a modelului cu datele.

MAE reprezintă o metrică folosită pentru a măsura acuratețea unui model de predicție. Aceasta reprezintă media absolutelor diferențelor dintre valorile observate și valorile prezise de model și se calculează cu ecuația (2.4).

MSE reprezintă o metrică de evaluare a unui model de predicție, care măsoară media pătratelor diferențelor dintre valorile observate și cele prezise de model. Este una dintre cele mai comune metrice utilizate pentru regresie și se calculează cu ecuația (2.5).

R^2 reprezintă o metrică statistică care indică proporția variabilității într-un set de date care este explicată de modelul de regresie. Este o măsură a cât de bine variază valorile prezise în funcție de valorile reale și se calculează cu ecuația (2.6).

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.4)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.5)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y})^2} \quad (2.6)$$

unde:

- n este numărul total de observații
- y_i sunt valorile observate
- \hat{y}_i sunt valorile prezise
- \bar{y} este media valorilor observate

Interpretare:

- MAE indică cât de departe, în medie, sunt predicțiile modelului de valorile reale. Cu cât MAE este mai mic, cu atât modelul este mai precis.
- MSE penalizează erorile mari mai mult decât pe cele mici deoarece erorile sunt pătrate. Un MSE mai mic indică un model mai precis.
- R^2 variază între 0 și 1.
- Dacă R^2 are valoarea 1 înseamnă că modelul explică perfect variabilitatea datelor.
- Dacă R^2 are valoarea 0 înseamnă că modelul nu explică deloc variabilitatea datelor.
- Un R^2 mai mare indică un model mai bun în explicarea variabilității datelor.

2.5.4. Limitările metodei alese

În cadrul dezvoltării aplicației noastre pentru estimarea compoziției corporale și monitorizarea sănătății utilizatorului, am ales să folosim algoritmi de învățare automată. Deși această metodă oferă numeroase avantaje, este important să identificăm și să înțelegem limitările asociate cu acestea. În această secțiune, vom discuta în detaliu despre limitările metodei alese din perspectiva algoritmilor de învățare automată.

• Dependența de calitatea datelor

Algoritmii de învățare automată sunt extrem de sensibili la calitatea datelor utilizate pentru antrenament și predicții. Datele de intrare incorecte, incomplete sau neactualizate pot conduce la rezultate inexacte și nesigure. În cazul aplicației noastre, măsurătorile corporale incorecte introduse de utilizatori pot afecta negativ acuratețea estimărilor privind compoziția corporală [27].

• Generalizarea pe diverse populații

Modelul de învățare automată antrenat pe un set de date specific poate să nu generalizeze bine la populații diferite. De exemplu, setul de date utilizat în program conține măsurători de la un grup specific de indivizi (252 de bărbați), ceea ce poate limita aplicabilitatea modelului la alte grupuri demografice (femei, copii, persoane cu diferite etnii). Acest lucru poate reduce precizia și utilitatea predicțiilor în aceste cazuri [27].

- **Complexitatea modelului**

Deși regresia liniară este un algoritm simplu și eficient, ea poate să nu fie suficient de puternică pentru a captura relații complexe între variabile. În cazul estimării compoziției corporale, pot exista interacțiuni complexe între diferite măsurători corporale care nu sunt capturate adecvat de un model liniar simplu. Utilizarea unor algoritmi mai avansați, cum ar fi rețelele neuronale, ar putea îmbunătăți acuratețea, dar ar necesita resurse computaționale mai mari și un efort mai mare pentru antrenament [27].

- **Dependența de densitatea corporală**

Calculul procentului de grăsime corporală se bazează pe densitatea corporală, care este influențată de mai mulți factori, inclusiv masa musculară și distribuția grăsimii. Aceasta poate duce la estimări inexacte pentru indivizii cu distribuții atipice ale masei corporale.

2.5.5. Compararea modelului cu alte tehnici

Conform literaturii de specialitate, mai multe tehnici de regresie și algoritmi de învățare automată au fost comparate în diverse studii. De exemplu, în studiul „*Body Fat Prediction using Various Regression Techniques*” [28], mai multe tehnici, inclusiv Random Forest, Gradient Boosting și Lasso Regression, au fost evaluate pe baza performanței lor în predicția BFP. Random Forest a demonstrat performanțe superioare în ceea ce privește RMSE și MAE, dar regresia liniară a oferit o interpretabilitate superioară și o complexitate computațională redusă.

Conform articolului „*Body Composition Methods: Comparisons and Interpretation*” de Dana L. Duren et al. [29] există mai multe metode de măsurare a grăsimii corporale, fiecare cu avantajele și dezavantajele sale:

- **Bioelectrical Impedance Analysis (BIA):**

Avantaje: Ușor de utilizat și neinvaziv. Poate fi efectuat rapid și cu echipament portabil.

Dezavantaje: Acuratețea depinde de starea de hidratare a individului. Predicțiile sunt mai puțin precise pentru indivizii obezi sau cei cu distribuții atipice ale masei corporale.

- **Dual-Energy X-ray Absorptiometry (DXA):**

Avantaje: Oferă măsurători precise ale masei osoase, musculare și a grăsimii corporale. Timp de scanare relativ scurt și expunere redusă la radiații.

Dezavantaje: Costisitor și necesită echipament specializat. Limitat de dimensiunea și greutatea subiectului, ceea ce poate exclude indivizii foarte obezi [29].

- **Hydrodensitometry (Underwater Weighing):**

Avantaje: Considerată una dintre cele mai precise metode pentru determinarea densității corporale.

Dezavantaje: Necesită echipament specializat și este incomodă pentru subiecți, în special pentru cei obezi sau copii. Subiecții trebuie să fie complet submersați și să-și țină respirația, ceea ce poate fi dificil pentru unii indivizi [29].

- **Computed Tomography (CT) și Magnetic Resonance Imaging (MRI):**

Avantaje: Oferă imagini detaliate și măsurători precise ale distribuției grăsimii corporale și a altor țesuturi.

Dezavantaje: Foarte costisitor și implică expunere la radiații (în cazul CT). Disponibilitatea limitată și accesibilitatea redusă pentru utilizarea de rutină [29].

- **Rețele neuronale și alte tehnici de învățare automată:** Studiul realizat de Zongwen Fan et al. [27] a investigat utilizarea rețelelor neuronale multilayer perceptron (MLP), mașinilor cu suport vectorial (SVM), pădurilor de decizie (RF) și XGBoost pentru predicția grăsimii corporale. Rezultatele au arătat că metodele de extracție a caracteristicilor, cum ar fi analiza componentelor principale (PCA) și analiza factorială (FA), pot îmbunătăți semnificativ performanța acestor modele, oferind predicții mai precise și mai stabile comparativ cu metodele tradiționale.

Deși metoda bazată pe învățarea automată oferă un instrument accesibil și neinvaziv pentru estimarea grăsimii corporale, are anumite limitări comparativ cu metodele mai avansate și specializate. Metodele directe și de referință, cum ar fi DXA și MRI, oferă măsurători mai precise și detaliate, dar sunt mai costisitoare și mai puțin accesibile. Alegerea metodei de măsurare a compoziției corporale depinde de contextul specific al utilizării, de resursele disponibile și de populația țintă [29].

2.5.6. Concluzii și avantaje ale abordării propuse

- Ușurința de utilizare:

Implementarea unui model de regresie liniară într-o aplicație Android este relativ simplă și eficientă din punct de vedere computațional.

- Precizie și interpretabilitate:

Modelul propus oferă predicții precise cu un coeficient de aproximativ $R^2 = 0.7$, ceea ce indică o bună potrivire a modelului. Coeficienții regresiei oferă o înțelegere directă a influenței greutateii și circumferinței abdomenului asupra procentului de grăsime corporală.

- Aplicabilitate practică:

Utilizatorii pot introduce doar două măsurători simple (greutatea și circumferința abdomenului) pentru a obține o estimare rapidă și precisă a BFP, ceea ce face modelul foarte practic pentru utilizarea zilnică.

2.6. Analiza performanțelor și descrierea componentelor hardware

Aplicația a fost dezvoltată pentru a funcționa eficient pe o gamă largă de dispozitive Android, utilizând resursele hardware și software disponibile în mod optim. În această secțiune, vom analiza performanțele aplicației, punând accent pe utilizarea resurselor și pe eficiența algoritmului de învățare automată.

2.6.1. Performanța aplicației pe diferite dispozitive

Aplicația este configurată să funcționeze pe dispozitive care au cel puțin Android 9 (API nivel 28). Astfel, aplicația poate rula pe orice dispozitiv care are Android 9 sau o versiune ulterioară. Testele de performanță au arătat că aplicația rulează eficient pe majoritatea dispozitivelor moderne, asigurând o experiență utilizator fluidă și receptivă.

2.6.2. Descrierea componentelor hardware

Pentru a asigura funcționarea corectă a aplicației, sunt necesare anumite componente hardware ale dispozitivului mobil. În această secțiune, vom descrie principalele componente hardware utilizate și rolul acestora în cadrul aplicației.

- **Senzorul de mișcare**

Senzorul de mișcare, cum ar fi accelerometrul și giroscopul, este utilizat pentru a monitoriza activitatea fizică a utilizatorilor, în special pentru contorizarea pașilor. Clasa StepCounterService folosește datele furnizate de acești senzori pentru a calcula numărul de pași parcurși

zilnic. Acești senzori sunt esențiali pentru funcționalitatea de monitorizare a activității fizice, asigurând acuratețea datelor colectate.

- **Modulul GPS**

Modulul GPS al dispozitivului este utilizat pentru a obține locația exactă a utilizatorilor. Acest lucru permite aplicației să afișeze locațiile sălilor de sport din apropiere folosind API-ul Google Maps. Clasa `FetchNearbyGymsTask` interacționează cu modulul GPS pentru a determina coordonatele utilizatorului și pentru a căuta locații relevante în proximitate.

- **Procesor și memorie**

Procesorul și memoria dispozitivului joacă un rol crucial în performanța generală a aplicației. Aplicația utilizează eficient resursele de procesare și memorie pentru a asigura o experiență utilizator fluidă. Algoritmul de regresie liniară, utilizat pentru estimarea compoziției corporale, este optimizat pentru a rula rapid și eficient pe dispozitive mobile.

Aplicația este proiectată să integreze fără probleme componentele hardware ale dispozitivelor Android, utilizând API-urile oferite de platforma Android. Această integrare permite colectarea eficientă a datelor și utilizarea optimă a resurselor hardware pentru a oferi o experiență utilizator de înaltă calitate.

2.7. Concluzii și direcții viitoare

Dezvoltarea acestei aplicații a demonstrat eficiența utilizării platformei Android și a serviciilor Firebase pentru construirea unei soluții scalabile și eficiente pentru monitorizarea sănătății și fitness-ului utilizatorilor. În viitor, aplicația poate fi extinsă pentru a include funcționalități suplimentare, cum ar fi:

Monitorizarea somnului: utilizarea senzorilor de mișcare și a algoritmilor de învățare automată pentru a monitoriza și analiza calitatea somnului utilizatorilor.

Integrarea cu alte dispozitive de sănătate: extinderea suportului pentru dispozitive de sănătate suplimentare, cum ar fi ceasuri inteligente și brățări de fitness, pentru o monitorizare mai cuprinzătoare a stării de sănătate.

Funcționalități sociale: adăugarea de funcționalități sociale care să permită utilizatorilor să-și împărtășească progresul și să se motiveze reciproc.

Capitolul 3. Implementarea aplicației

3.1. Descrierea generală a implementării

Aplicația pentru estimarea procentajului de grăsime și monitorizarea activităților fizice a fost dezvoltată pentru a oferi utilizatorilor o soluție integrată și ușor de utilizat pentru monitorizarea sănătății. Tehnologiile utilizate includ Java pentru dezvoltarea aplicației Android, Python pentru dezvoltarea modelului de învățare automată și Firebase pentru gestionarea autentificării și stocării datelor. Proiectul nostru utilizează, pentru o parte a aplicației, arhitectura MVVM (Model-View-ViewModel) care separă logica de prezentare de logica de afaceri și de accesul la date, asigurând astfel o dezvoltare modulară.

Aplicația este un instrument complex și complet pentru monitorizarea și îmbunătățirea stării de sănătate a utilizatorilor săi. Dezvoltată pentru platforma Android, aplicația integrează o serie de funcționalități esențiale, cum ar fi calculul BMI (Indicele de Masă Corporală), estimarea procentului de grăsime corporală, monitorizarea numărului de pași zilnici și localizarea sălilor de fitness din apropiere. Implementarea acestei aplicații a implicat utilizarea Android SDK, Firebase pentru autentificare și stocare în timp real, și API-urilor Google pentru servicii de localizare și hărți.

Programul este structurat în module interconectate, fiecare responsabil pentru o anumită funcționalitate. Modulul de calcul al BMI-ului permite utilizatorilor să își introducă datele personale pentru a obține instantaneu valoarea BMI-ului și interpretarea acesteia. Modulul de estimare a procentului de grăsime corporală utilizează măsurători precise și algoritmi de învățare automată pentru a oferi utilizatorilor o estimare exactă a nivelului de grăsime corporală. Modulul de monitorizare a pașilor folosește senzorii dispozitivului pentru a contoriza pașii zilnici și pentru a motiva utilizatorii să atingă obiectivele zilnice de activitate fizică.

Fiecare modul este conceput pentru a interacționa eficient cu celelalte, asigurând o experiență integrată și ușor de utilizat pentru utilizatori. De exemplu, datele colectate de la senzorii de mișcare sunt stocate în Firebase, permițând accesul în timp real de pe orice dispozitiv. În plus, aplicația utilizează Google Maps și API-ul Google Places pentru a oferi informații actualizate despre sălile de fitness din apropiere, îmbunătățind astfel utilitatea și funcționalitatea generală a aplicației.

3.2. Probleme speciale/dificultăți întâmpinate și modalități de rezolvare

Implementarea aplicației a adus o serie de provocări tehnice care au necesitat soluții inovatoare și eficiente. Iată câteva dintre principalele probleme întâmpinate și modalitățile de rezolvare adoptate:

3.2.1. Probleme de integrare a Firebase

Integrarea Firebase Authentication și Firebase Realtime Database a prezentat mai multe provocări tehnice. Unul dintre principalele obstacole a fost asigurarea sincronizării datelor în timp real între multiple dispozitive. De asemenea, gestionarea erorilor de autentificare și menținerea sesiunilor utilizatorilor au necesitat soluții eficiente.

Pentru a rezolva aceste probleme, s-au utilizat callback-uri și observatoare pentru a monitoriza schimbările de date și starea autentificării. Callback-urile ne-au permis să reacționăm prompt la evenimentele de autentificare și modificările datelor, asigurând astfel o experiență utilizator fără întreruperi. Observatoarele au fost utilizate pentru a urmări modificările în timp real ale datelor stocate în Firebase Realtime Database, permițând aplicației să sincronizeze rapid informațiile între toate dispozitivele utilizatorilor.

3.2.2. Calculul și afișarea BMI și a grăsimii corporale

Utilizarea formulelor de calcul și integrarea acestora cu interfața a necesitat atenție la detalii pentru a asigura acuratețea și afișarea corectă a rezultatelor. Am utilizat formule standardizate pentru calculul BMI și un model de regresie liniară pentru predicția grăsimii corporale. Am calibrat coeficienții modelului de regresie pentru a îmbunătăți acuratețea predicțiilor, folosind tehnici de preprocesare pentru a elimina outlierii.

3.2.3. Localizarea sălilor de fitness

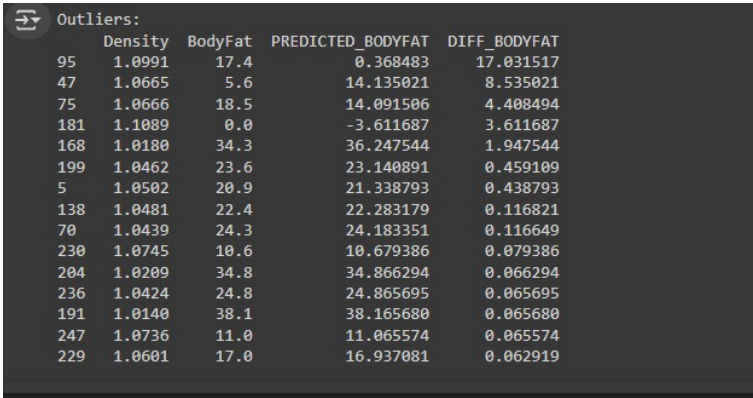
Integrarea Google Maps și Google Places API pentru a permite utilizatorilor să găsească săli de fitness în apropierea lor a fost o provocare, în special gestionarea permisiunilor de locație și optimizarea cererilor pentru a minimiza consumul de resurse.

Am folosit Google Maps API pentru a afișa harta și Google Places API pentru a căuta săli de fitness în apropiere. Am gestionat permisiunile de locație folosind API-urile Android, solicitând utilizatorilor permisiunile necesare la momentul potrivit și optimizând cererile pentru a reduce consumul de date și resurse. Markerii de pe hartă au fost colorați pentru a distinge mai bine locațiile relevante.

3.2.4. Performanța modelului de învățare automată

Inițial, modelul de regresie liniară dezvoltat în Python nu a oferit performanțele așteptate în ceea ce privește precizia estimărilor compoziției corporale. Aceasta s-a datorat prezenței outlier-ilor și a datelor zgomotoase în setul de date utilizat pentru antrenarea modelului.

Pentru a îmbunătăți performanța modelului, am curățat setul de date eliminând outlier-ii și normalizând datele. De asemenea, am testat diferiți algoritmi, tehnici, metode pentru ca în final să alegem soluția potrivită în cazul nostru. Am comparat grăsimea corporală estimată dedusă din densitate și grăsimea corporală de intrare și am verificat care dintre ele este mai rezonabilă. În acest fel, am putea detecta unele potențiale valori aberante.



	Density	BodyFat	PREDICTED_BODYFAT	DIFF_BODYFAT
95	1.0991	17.4	0.368483	17.031517
47	1.0665	5.6	14.135021	8.535021
75	1.0666	18.5	14.091506	4.408494
181	1.1089	0.0	-3.611687	3.611687
168	1.0180	34.3	36.247544	1.947544
199	1.0462	23.6	23.140891	0.459109
5	1.0502	20.9	21.338793	0.438793
138	1.0481	22.4	22.283179	0.116821
70	1.0439	24.3	24.183351	0.116649
230	1.0745	10.6	10.679386	0.079386
204	1.0209	34.8	34.866294	0.066294
236	1.0424	24.8	24.865695	0.065695
191	1.0140	38.1	38.165680	0.065680
247	1.0736	11.0	11.065574	0.065574
229	1.0601	17.0	16.937081	0.062919

Figura 3.1. Outliers

3.2.5. Probleme de compatibilitate a dispozitivelor

Diversitatea dispozitivelor Android, cu diferite versiuni de sistem de operare și specificații hardware, a reprezentat o altă provocare majoră. Asigurarea compatibilității aplicației pe o gamă largă de dispozitive a necesitat teste extinse și optimizări pentru a garanta performanța și funcționalitatea corectă.

Pentru a aborda aceste probleme, am utilizat emulatoare și dispozitive fizice (cu precădere dispozitive Samsung) cu diferite specificații pentru a testa aplicația în diverse scenarii.

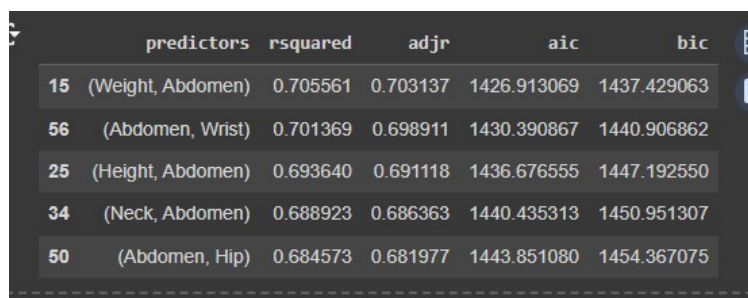
3.2.6. Contorizarea pașilor

Asigurarea unei contorizări precise a pașilor a fost o provocare majoră. A fost necesară o gestionare atentă a datelor furnizate de senzorii dispozitivului pentru a evita erorile și inconsistențele.

3.3. Idei originale, soluții noi

3.3.1. Utilizarea algoritmilor de învățare automată pentru estimarea compoziției corporale

Aplicația noastră utilizează un model de învățare automată pentru a estima compoziția corporală pe baza măsurătorilor antropometrice. Acest model de regresie liniară a fost dezvoltat în Python și integrat în aplicația Android. Modelul utilizează date precum circumferințele corporale, vârsta, greutatea și înălțimea pentru a oferi estimări precise ale procentului de grăsime corporală. În imaginea 3.2 este prezentat un tabel cu rezultatele unor modele de regresie liniară care utilizează diferiți predictori pentru a estima procentul de grăsime corporală. Am ales să utilizăm doar doi predictori pentru a ușura interacțiunea utilizatorului și a nu fi nevoie de multe măsurători. Totodată, folosirea mai multor predictori nu îmbunătățeau îndeajuns modelul încât să fie luate în considerare. Putem concluziona că variabilele greutate și abdomen sunt cei mai buni predictori pentru estimarea procentului de grăsime corporală în comparație cu celelalte combinații de variabile testate.



	predictors	rsquared	adjr	aic	bic
15	(Weight, Abdomen)	0.705561	0.703137	1426.913069	1437.429063
56	(Abdomen, Wrist)	0.701369	0.698911	1430.390867	1440.906862
25	(Height, Abdomen)	0.693640	0.691118	1436.676555	1447.192550
34	(Neck, Abdomen)	0.688923	0.686363	1440.435313	1450.951307
50	(Abdomen, Hip)	0.684573	0.681977	1443.851080	1454.367075

Figura 3.2. Compararea performanțelor diferitelor modele de regresie liniară

Avantajul acestei abordări este că utilizatorii pot beneficia de o metodă accesibilă pentru monitorizarea compoziției corporale, evitând necesitatea utilizării echipamentelor costisitoare.

3.3.2. Monitorizarea activității fizice prin senzori

Pentru monitorizarea activității fizice, aplicația utilizează senzorii încorporați ai dispozitivului mobil, cum ar fi accelerometrul și giroscopul. Acestea sunt utilizate pentru a monitoriza numărul de pași și mișcările utilizatorilor. Datele colectate sunt stocate în Firebase Realtime Database și sunt utilizate pentru a genera grafice și statistici detaliate despre activitatea fizică a utilizatorilor.

3.3.3. Calculul necesarului caloric și integrarea NewsAPI

Am implementat o funcționalitate care permite utilizatorilor să își determine necesarul caloric zilnic în funcție de obiectivul lor (pierdere în greutate, menținere, creștere în greutate) și nivelul de activitate fizică. Aceste informații sunt apoi utilizate pentru a oferi recomandări personalizate și articole relevante.

Pentru a furniza utilizatorilor articole de sănătate actualizate și relevante, am integrat NewsAPI și am implementat un filtru pentru categorisirea articolelor în funcție de cuvintele cheie.

3.4. Funcționarea sistemului

Aplicația este structurată în mai multe module, fiecare având roluri și responsabilități specifice. Aceste module sunt interconectate și colaborează pentru a oferi o experiență utilizator

completă și integrată.

Funcționarea sistemului este prezentată printr-o serie de capturi de ecran care ilustrează principalele funcționalități ale aplicației. Aceste capturi sunt însoțite de descrieri detaliate pentru a oferi o imagine completă a modului în care utilizatorii pot interacționa cu aplicația.

Ecranul de întâmpinare (MainActivity)

Această pagină reprezintă prima pagină de interacționare a utilizatorului cu aplicația. În cadrul acesteia, utilizatorul poate alege între autentificare în cazul în care are deja un cont creat sau înregistrare în cazul în care nu și-a creat încă un cont.

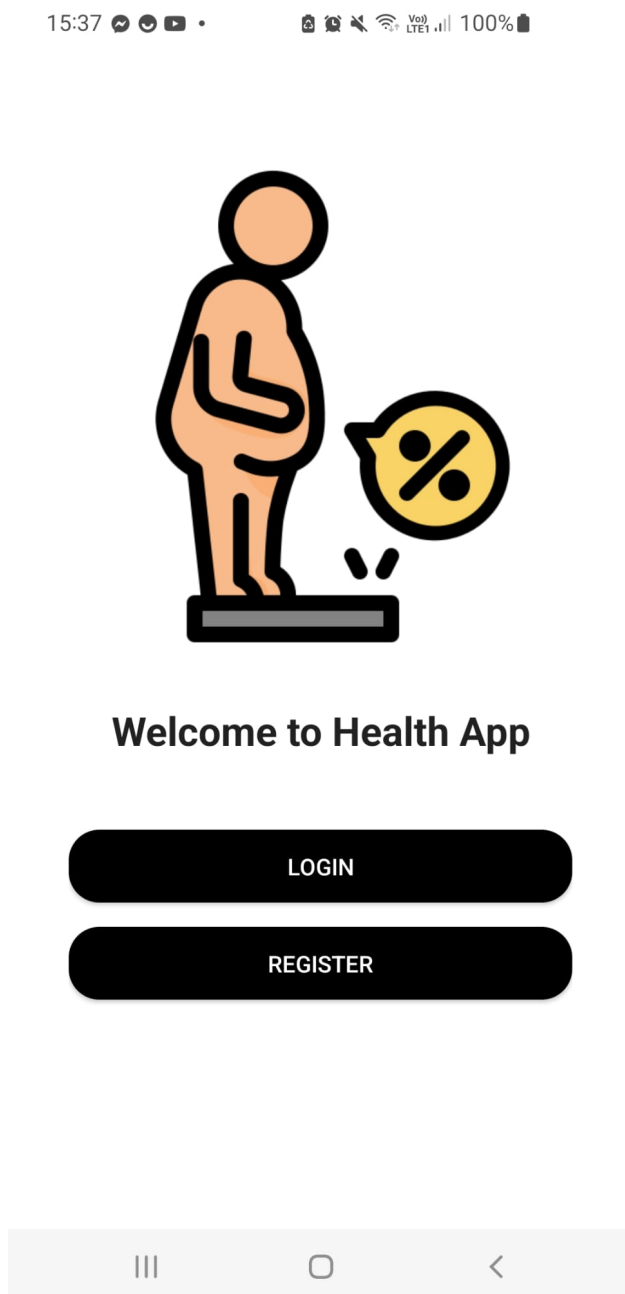


Figura 3.3. Prima pagină

Ecranul de autentificare și înregistrare

Autentificarea și înregistrarea utilizatorilor sunt esențiale pentru personalizarea experienței și stocarea datelor individuale. Ecranele de autentificare (LoginActivity) și înregistrare (RegisterActivity) permit utilizatorilor să se logheze sau să își creeze un cont nou.

15:39 100%

Welcome Back

Email or Username

Password [Forgot Password?](#)

SIGN IN

Logged out

III □ <

Figura 3.4. Pagina de logare

15:38 100%

Register Here

Full Name*
Enter your full name

Email*
Enter your email

Date of Birth (dd/mm/yyyy)*
Select your date of birth (dd/mm/yyyy)

Gender*
☐ Female ☐ Male

Mobile*
Enter your mobile number

Password*
Enter your password

Confirm Password*
Confirm your password

III □ <

Figura 3.5. Pagina de creare cont

Ecranele de vizualizare și gestionare a profilului

În cadrul acestora utilizatorii au posibilitatea de a vizualiza și actualiza informațiile asociate contului printre care: informațiile personale, schimbarea parolei, mail-ului, a imaginii de profil, ștergerea contului și deconectarea.

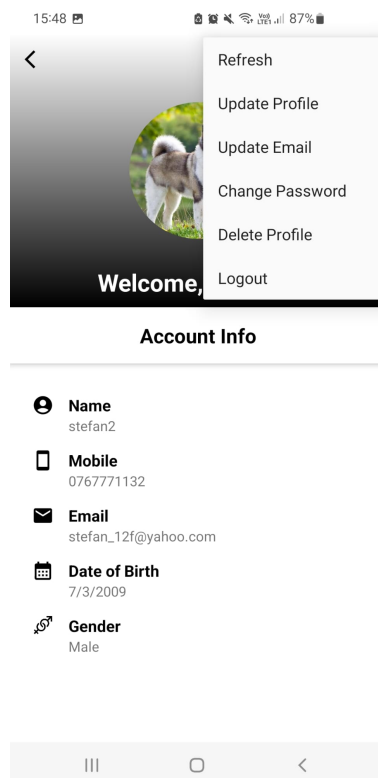


Figura 3.6. Ecranul de vizualizare a profilului

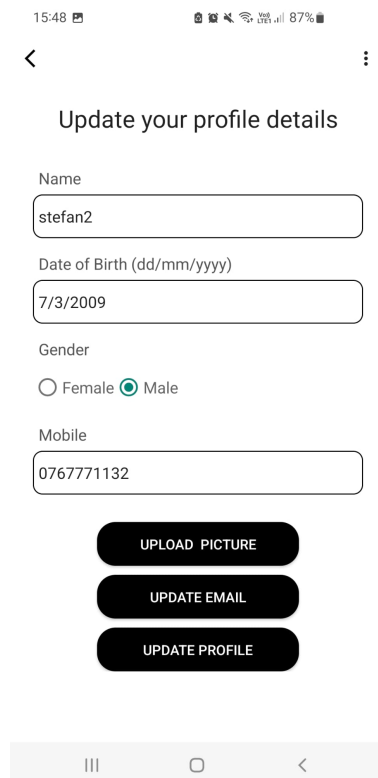


Figura 3.7. Ecranul de update al profilului

Ecranul de sumar al utilizatorului

Ecranul principal al aplicației oferă un sumar al informațiilor de sănătate ale utilizatorului, inclusiv BMI, procentul de grăsime corporală, necesarul caloric și numărul de pași efectuați zilnic. Utilizatorii pot naviga ușor între funcționalități prin intermediul meniului de navigare.

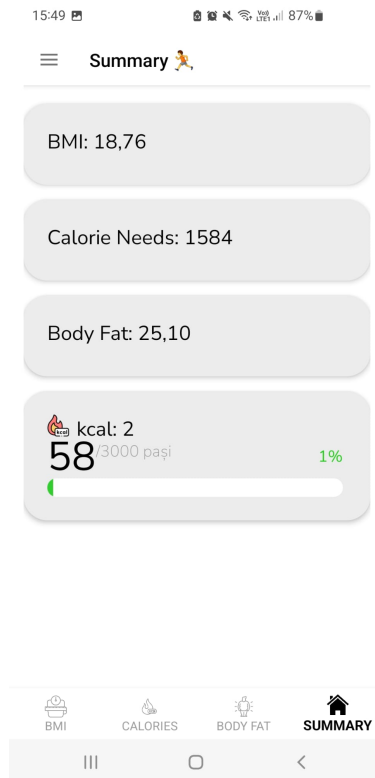


Figura 3.8. Ecranul de sumar al utilizatorului

Ecranul de calcul al BMI-ului

Acest ecran permite utilizatorilor să își calculeze BMI-ul pe baza înălțimii și greutatei lor. Rezultatul este afișat instantaneu, împreună cu o interpretare a valorii BMI care indică dacă utilizatorul se află în intervalul normal, subponderal sau supraponderal.

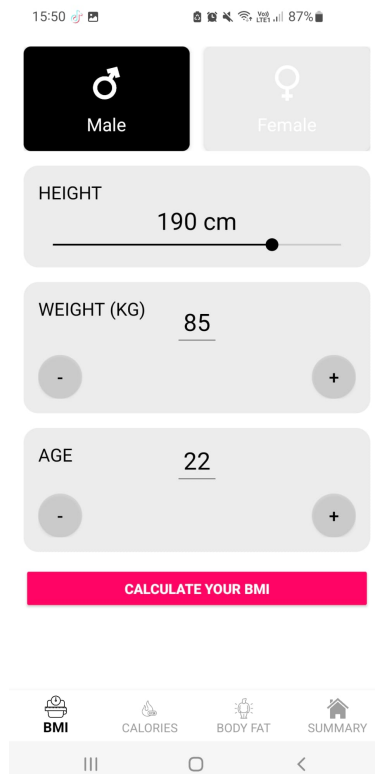


Figura 3.9. Ecranul de calcul al BMI-ului

Ecranul de estimare a procentului de grăsime corporală și de localizare a sălilor de fitness

Utilizatorii pot introduce măsurători ale abdomenului și greutateii pentru a obține o estimare a procentului de grăsime corporală. Algoritmul utilizat pentru această estimare este bazat pe un model de regresie liniară. Totodată în cadrul acestui ecran poate fi vizualizată harta cu locația sălilor de fitness.

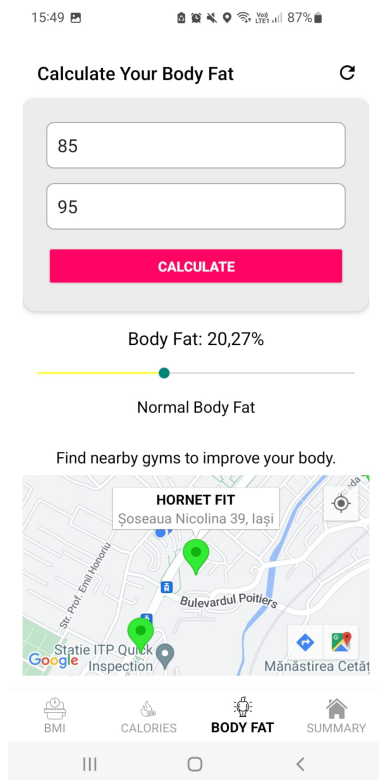


Figura 3.10. Ecranul de calcul al procentului de grăsime

```

1  handler.post(() -> {
2      try {
3          JSONObject jsonObject = new JSONObject(finalResponse);
4          JSONArray resultsArray = jsonObject.getJSONArray("results");
5
6          for (int i = 0; i < resultsArray.length(); i++) {
7              JSONObject placeObject = resultsArray.getJSONObject(i);
8              String name = placeObject.getString("name");
9              String address = placeObject.getString("vicinity");
10             JSONObject location =
11                 ↳ placeObject.getJSONObject("geometry").getJSONObject("location");
12             double lat = location.getDouble("lat");
13             double lng = location.getDouble("lng");
14             LatLng gymLocation = new LatLng(lat, lng);
15
16             mMap.addMarker(new MarkerOptions()
17                 .position(gymLocation)
18                 .title(name)
19                 .snippet(address)
20                 ↳ .icon(BitmapDescriptorFactory.defaultMarker(BitmapDescrip

```

```

20         }
21     } catch (Exception e) {
22         Log.e("FetchNearbyGymsTask", "Error parsing JSON: ", e);
23     }
24 });

```

Listing 3.1. Actualizarea UI pentru afișarea sălilor de fitness apropiate

Codul din 3.1 reprezintă un handler (permite actualizarea interfeței utilizatorului pe firul principal, asigurând că orice modificare a UI-ului este făcută în mod sigur) din cadrul clasei `FetchNearbyGyms` care transformă răspunsul HTTP într-un obiect JSON care apoi este „parsificat” pentru a extrage informațiile necesare (nume, adresă, locație).

Ecranele de monitorizare a pașilor

Acest ecran afișează numărul de pași efectuați zilnic și oferă statistici detaliate. Utilizatorii pot seta obiective zilnice și pot vizualiza progresul lor pe termen lung. Graficele interactive și statisticile detaliate ajută utilizatorii să își mențină motivația și să își atingă obiectivele. Pentru afișarea graficelor și vizualizarea datelor într-un mod intuitiv și atractiv, s-a utilizat biblioteca `MPAndroidChart` care oferă componente pentru crearea graficelor și diagramelor.

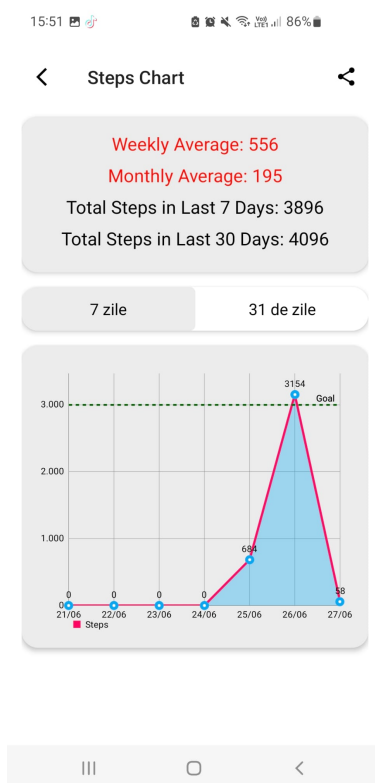


Figura 3.11. Ecranul de vizualizare detaliată a pașilor

Ecranul de calcul al necesarului caloric și de articole de sănătate

Acest ecran afișează necesarul caloric al utilizatorului în funcție de greutate, înălțime, vârstă, obiectiv (slăbire/menținere/îngrășare) și de activitatea fizică. În funcție de obiectivul ales vor fi afișate mai apoi articole relevante.

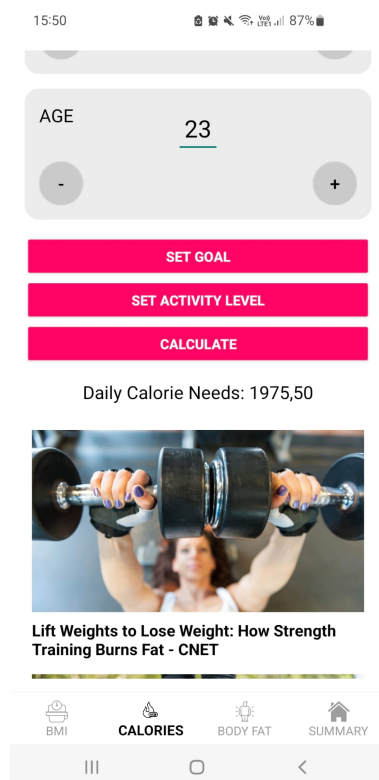


Figura 3.12. Ecranul de vizualizare a necesarului caloric

3.5. Comunicarea cu alte sisteme și salvarea/ stocarea informațiilor

Aplicația comunică cu mai multe sisteme externe pentru a oferi funcționalitățile sale:

- **Firebase:** Pentru autentificarea utilizatorilor, folosim Firebase Authentication. Aceasta permite utilizatorilor să se înregistreze și să se autentifice utilizând adrese de e-mail și parole. Firebase gestionează toate detaliile legate de autentificare, inclusiv stocarea sigură a parolelor și gestionarea sesiunilor de utilizator. Un avantaj major al utilizării Firebase Authentication este integrarea ușoară cu alte servicii Firebase, precum Realtime Database și Firebase Storage. Datele utilizatorilor, cum ar fi informațiile personale, datele de sănătate și activitatea fizică, sunt stocate în Firebase Realtime Database. Structura bazei de date este organizată astfel încât fiecare utilizator are un nod dedicat sub "Registered Users", identificat prin UID-ul (User ID) unic generat de Firebase Authentication. Imaginile de profil ale utilizatorilor sunt stocate în Firebase Storage. URL-ul imaginii stocate este salvat în Firebase Realtime Database, astfel încât poate fi accesat și afișat în aplicație. Structura URL-ului este similară cu cea din exemplul de mai sus. Pentru a proteja datele utilizatorilor, am implementat reguli de securitate atât pentru Firebase Realtime Database, cât și pentru Firebase Storage. Aceste reguli se asigură că doar utilizatorii autentificați au acces la datele lor și că aceștia nu pot accesa datele altor utilizatori.



Figura 3.13. Structura datelor stocate în Firebase Realtime Database


```

1      private void saveBodyFatToFirebase(double bodyFat) {
2      FirebaseUser user =
        ↳ FirebaseAuth.getInstance().getCurrentUser();
3      if (user != null) {
4          DatabaseReference ref =
            ↳ FirebaseDatabase.getInstance().getReference("Registered
            ↳ Users")
5              .child(user.getId()).child("bodyFat");
6          ref.setValue(bodyFat);
7      }
8  }

```

Listing 3.2. Exemplu de cod pentru salvarea datelor în Firebase:

- **Google Maps API:** Utilizat pentru a găsi și afișa sălile de fitness din apropierea utilizatorului. API-ul Google Maps oferă acces la o bază de date vastă de locuri și permite personalizarea afișării hărților în aplicație.
- **NewsAPI:** Folosit pentru a obține articole de sănătate relevante care sunt afișate utilizatorilor în funcție de interesele lor. Acesta permite accesul la o varietate de surse de știri și articole de sănătate, oferind utilizatorilor informații actualizate și relevante.

Pentru comunicarea cu aceste API-uri externe și gestionarea cererilor HTTP, s-au utilizat bibliotecile Retrofit și OkHttp. Retrofit facilitează realizarea cererilor HTTP și conversia răspunsurilor JSON în obiecte Java. Acest lucru este esențial pentru interacțiunea cu NewsAPI. OkHttp ajută la realizarea cererilor și gestionarea conexiunilor HTTP. Este utilizat în spatele Retrofit.

3.6. Interfața cu utilizatorul

Interfața cu utilizatorul (UI) este o componentă esențială a aplicației, deoarece aceasta determină modul în care utilizatorii interacționează cu funcționalitățile oferite. Această componentă a proiectului a fost mult usurată de Android Studio, un mediu de dezvoltare integrat (IDE) care oferă instrumente puternice pentru crearea de interfețe grafice atractive și responsive.

Biblioteci pentru interfața cu utilizatorul

Pentru a crea o interfață de utilizator modernă și atractivă, s-au folosit diverse biblioteci care facilitează crearea și gestionarea elementelor UI.

AndroidX: Oferă componente pentru dezvoltarea interfeței cu utilizatorul, asigurând compatibilitatea și funcționalitatea pe diferite versiuni de Android.

Material Design: Asigură respectarea principiilor de design modern, oferind componente vizuale și interacțiuni estetice.

Glide și Picasso: Biblioteci utilizate pentru încărcarea și gestionarea eficientă a imaginilor. Acestea ajută la afișarea imaginilor de profil și a altor resurse vizuale în aplicație.

Utilizarea fișierelor XML: În Android, interfața grafică este definită utilizând fișiere XML. Fiecare activitate și fragment din aplicație are un fișier XML asociat care descrie aspectul vizual și structura layout-ului. Aceste fișiere specifică elementele UI precum butoanele, textul, imaginile și alți indicatori vizuali, precum și aranjamentele lor în cadrul ecranului.

Fișierele XML sunt stocate în directorul `res/layout` al proiectului și sunt referențiate în codul Java sau Kotlin al aplicației. Acest lucru permite o separare clară între logica de afaceri și designul interfeței, facilitând astfel dezvoltarea și întreținerea aplicației.

Tipuri de layout-uri utilizate Pentru a crea interfața aplicației, am utilizat diverse tipuri de layout-uri disponibile în Android, fiecare având caracteristici specifice care permit aranjarea elementelor UI în moduri diferite. Printre acestea se numără:

- **LinearLayout:** Permite aranjarea elementelor UI în linie orizontală sau verticală.
- **RelativeLayout:** Permite poziționarea elementelor UI în raport unul față de celălalt sau față de marginile containerului.
- **ConstraintLayout:** Oferă un sistem flexibil de constrângeri pentru aranjarea elementelor UI într-un mod mai complex și responsive.

3.6.1. Integrarea componentelor UI

Fiecare componentă UI, precum butoanele, câmpurile de text și imaginile, este definită în fișierele XML și configurată prin intermediul atributelor specifice. De exemplu, un buton poate avea attribute care definesc textul afișat, dimensiunea, culoarea și marginile.

Interacțiunea cu utilizatorul: Interacțiunea cu utilizatorul este gestionată prin evenimente și callback-uri. De exemplu, atunci când un utilizator apasă un buton, este declanșat un eveniment `onClick` care poate fi gestionat în codul Java sau Kotlin asociat activității sau fragmentului respectiv.

Personalizarea și teme: Android Studio permite personalizarea interfeței utilizatorului prin utilizarea temelor și stilurilor. Acestea sunt definite în fișierele XML din directorul `res/values`, cum ar fi `styles.xml`. Utilizând teme, putem aplica un set consistent de attribute vizuale, cum ar fi culori, fonturi și dimensiuni, la nivelul întregii aplicații.

Pentru o prezentare detaliată a diferitelor ecrane și interacțiuni din cadrul aplicației, vă rugăm să consultați capitolul 3.4, unde sunt incluse capturi de ecran și descrieri detaliate ale funcționalităților principale.

Capitolul 4. Testarea aplicației și rezultate experimentale

4.1. Punerea în funcțiune/ lansarea aplicației, elemente de configurare sau instalare

Pentru a pune în funcțiune aplicația este necesară instalarea acesteia pe un dispozitiv Android. După instalare, utilizatorul trebuie să se autentifice folosind email-ul și parola pentru a accesa funcționalitățile aplicației. Pentru utilizatorii noi, aplicația oferă o opțiune de înregistrare, unde aceștia pot crea un cont nou. Odată autentificat, utilizatorul poate începe să folosească toate funcționalitățile oferite de aplicație. Pentru a pune aplicația în funcțiune există mai multe metode:

1. Generarea și Instalarea unui APK:

- **Descărcarea fișierului APK:** utilizatorul va descărca fișierul APK de pe o sursă furnizată de dezvoltator.
- **Permisuni de instalare:** utilizatorul trebuie să permită instalarea aplicațiilor din surse necunoscute în setările dispozitivului Android.
- **Instalarea aplicației:** pentru realizarea acestui pas, utilizatorul trebuie să deschidă fișierul APK și să urmeze instrucțiunile de instalare.

2. Instalarea Aplicației pe un Dispozitiv Fizic prin USB

- **Activarea modului developer și debugging prin USB**
- **Conectarea dispozitivului la PC:** se va conecta dispozitivul Android la computer folosind un cablu USB.
- **Autorizarea computerului pentru Debugging:** pe dispozitivul Android, acceptă promptul pentru a autoriza computerul pentru debugging.
- **Rularea aplicației:** în Android Studio, apasăm pe butonul Run. Se va selecta dispozitivul fizic din lista de dispozitive disponibile. Android Studio va compila aplicația și o va instala pe dispozitivul conectat.

3. Rularea aplicației pe un emulator Android

- **Configurarea emulatorului:** în Android Studio, accesăm Tools > AVD Manager. Apoi, vom crea un nou dispozitiv virtual Android (AVD) selectând un dispozitiv din lista disponibilă și configurându-l după preferințe.
- **Rularea aplicației:** deschidem proiectul în Android Studio. Se apasă pe butonul Run (sau utilizăm comanda rapidă Shift+F10). În final, selectăm emulatorul configurat anterior. Android Studio va compila aplicația și o va rula pe emulator.

4.2. Testarea sistemului (hardware/ software)

Procesul de testare a sistemului a fost necesar pentru a garanta că aplicația funcționează și este de calitate pe o gamă largă de dispozitive și versiuni de Android. Pentru a ne asigura că aplicația funcționează corect indiferent de versiunea de hardware sau software utilizată, am testat-o pe o varietate de dispozitive.

Testarea hardware: A inclus verificarea compatibilității aplicației cu o varietate de modele de telefoane, asigurându-ne că senzorii necesari, cum ar fi senzorii de pași, funcționează corect pe toate dispozitivele testate. Pentru a realiza aceste teste, am folosit atât emulatoare, cât și dispozitive fizice.

Testarea software-ului a inclus verificarea corectitudinii funcționalităților principale ale aplicației:

- Înregistrarea și autentificarea: testarea autentificării și înregistrării utilizatorilor folosind Firebase Authentication
- Calculul indicelui de masă corpului (BMI): verificarea corectitudinii formulelor utilizate pentru calculul BMI.
- Estimarea grăsimii corporale: estimările sunt verificate folosind modelul de regresie liniară.
- Procesul de monitorizare: verifică funcționarea senzorului de pași și corectitudinea datelor stocate în Firebase. S-au comparat rezultatele din aplicația dezvoltată cu rezultatele altor aplicații consacrate.
- Localizare săli de fitness: testarea integrării cu Google Maps și Google Places API pentru a afișa corect sălile de fitness din apropiere.

4.3. Aspecte legate de încărcarea procesorului, memoriei, limitări în ce privește transmiterea datelor/ comunicarea

Pentru evaluarea performanței aplicației au fost utilizate instrumentele de profilare din Android Studio, care oferă informații detaliate despre utilizarea CPU și memoriei. Profilul de performanță a fost realizat pe mai multe activități critice ale aplicației, cum ar fi autentificarea, încărcarea datelor de profil, calculul necesarului caloric, monitorizarea pașilor, etc. Acest lucru este esențial pentru a asigura o experiență fluidă utilizatorilor și pentru a evita consumul excesiv de resurse care ar putea duce la încetinirea dispozitivului sau la descărcarea rapidă a bateriei.

4.3.1. Utilizarea CPU

Profilul CPU al aplicației, așa cum este prezentat în figura 4.1, indică faptul că aplicația utilizează în medie sub 30% din capacitatea procesorului în timpul funcționării normale. Acest lucru include procesarea datelor pentru monitorizarea pașilor, calculul necesarului caloric și gestionarea interacțiunilor utilizatorului. Totodată se poate observa că utilizarea procesorului de către program atinge *vârful* (55%) graficului în momentul accesării fragmentului din cadrul clasei SummaryActivity pentru calcularea procentajului de grăsime, fapt datorat de folosirea API-ului Google pentru încărcarea hărții cu sălile de fitness.

Analiza utilizării CPU arată că aplicația menține o utilizare relativ constantă a procesorului, cu vârfuri ocazionale în timpul sarcinilor intensive. Cu toate acestea există spațiu pentru optimizare, în special pentru a reduce vârfurile și a asigura o experiență mai lină pentru utilizatori.

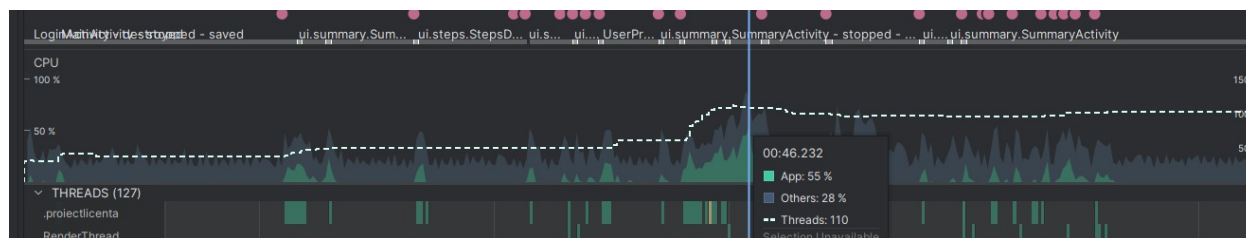


Figura 4.1. Utilizarea CPU de către aplicație

4.3.2. Utilizarea Memoriei

Profilul memoriei, așa cum se poate observa în imaginea 4.2, arată o utilizare constantă a memoriei pe măsură ce diferite activități sunt inițiate și gestionate. În general, aplicația utilizează

aproximativ 133.9 MB de memorie, ceea ce este considerat acceptabil pentru o aplicație de acest tip. De asemenea, graficul atinge valorile maxime în momentul accesării fragmentului cu harta sălilor.

Datele indică o alocare stabilă, cu o creștere treptată pe măsură ce mai multe activități sunt desfășurate. Cu toate acestea, pentru a îmbunătăți performanța, ar putea fi necesară optimizarea managementului memoriei pentru a reduce creșterile abrupte și a preveni eventualele scurgeri de memorie.

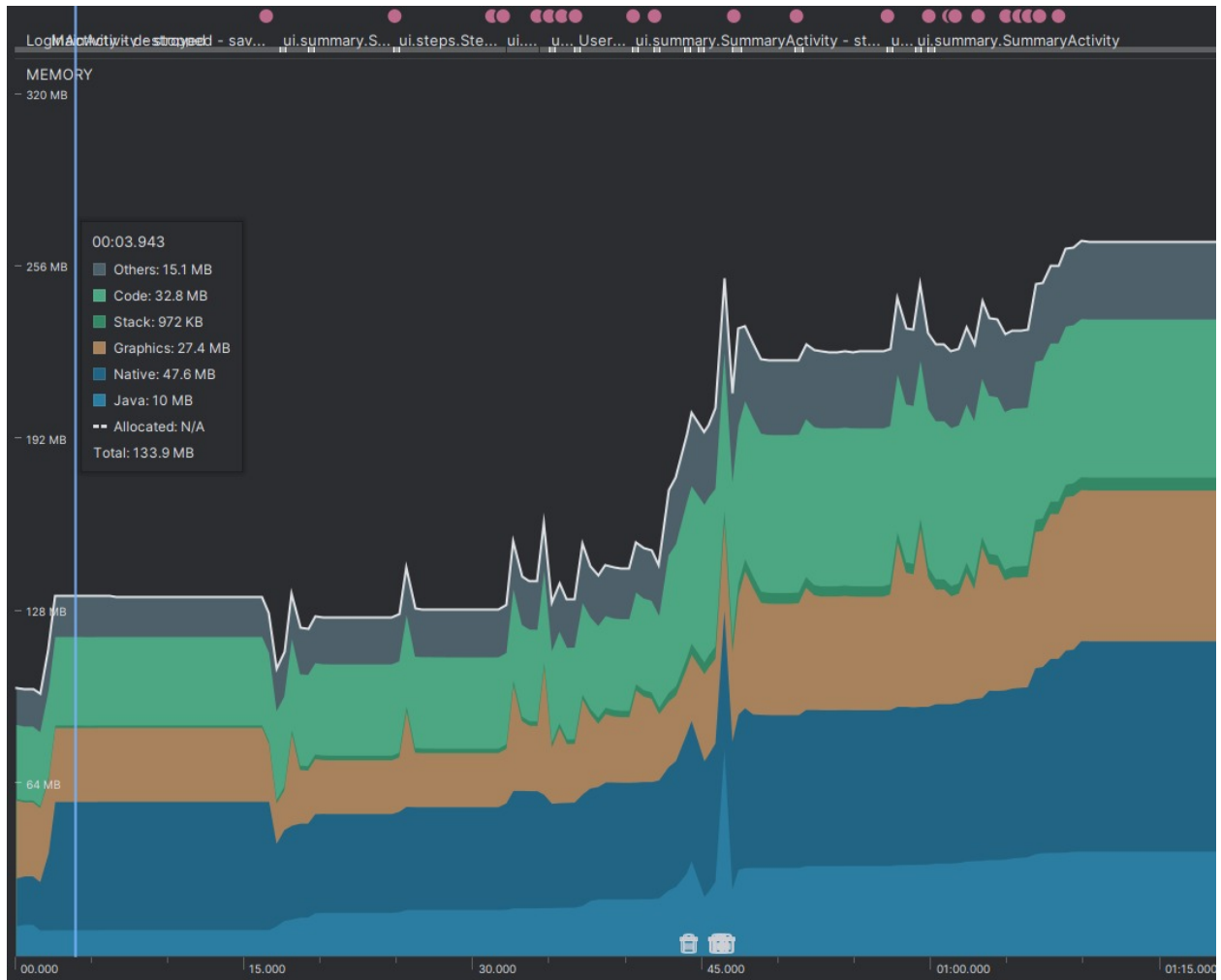


Figura 4.2. Utilizarea memoriei de către aplicație

4.4. Date de test/ metrici/ benchmarks

A fost efectuată o evaluare a modelului de învățare automată pentru a determina procentul de grăsime corporală. Acest proces a implicat utilizarea mai multor metrici de performanță. În tabelul 4.1 sunt prezentate rezultatele diferitelor modele de învățare automată aplicate pe setul de date avut.

Model	Mean Absolute Error	Mean Squared Error	R ² Score
Linear Regression	3.296760	17.759100	0.705084
Decision Tree	4.568000	30.807733	0.488392
Random Forest	3.405160	17.155565	0.715106
Gradient Boosting	3.356719	18.375723	0.694844
K-Nearest Neighbors	4.394933	25.443963	0.577465

Tabelul 4.1. Performanța diferitelor modele de învățare automată

Figura 4.3 prezintă comparativ performanța MAE (Mean Absolute Error) pentru diferite modele de învățare automată, evidențiind acuratețea predicțiilor fiecărui model.

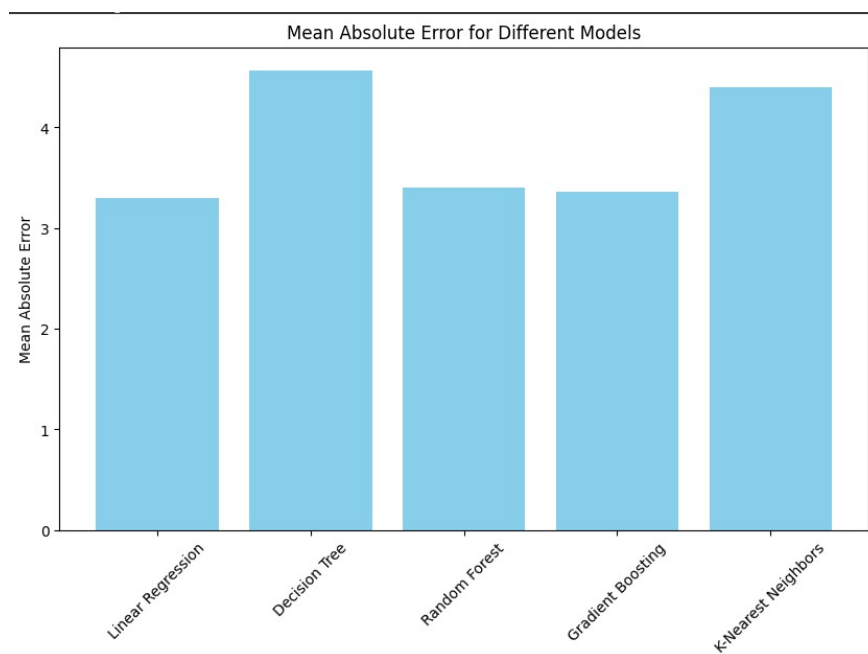


Figura 4.3. Mean Absolute Error pentru diferite modele

De asemenea, figura 4.4 prezintă MSE (Mean Squared Error) pentru aceleași modele, oferind o perspectivă asupra variabilității erorilor predicțiilor.

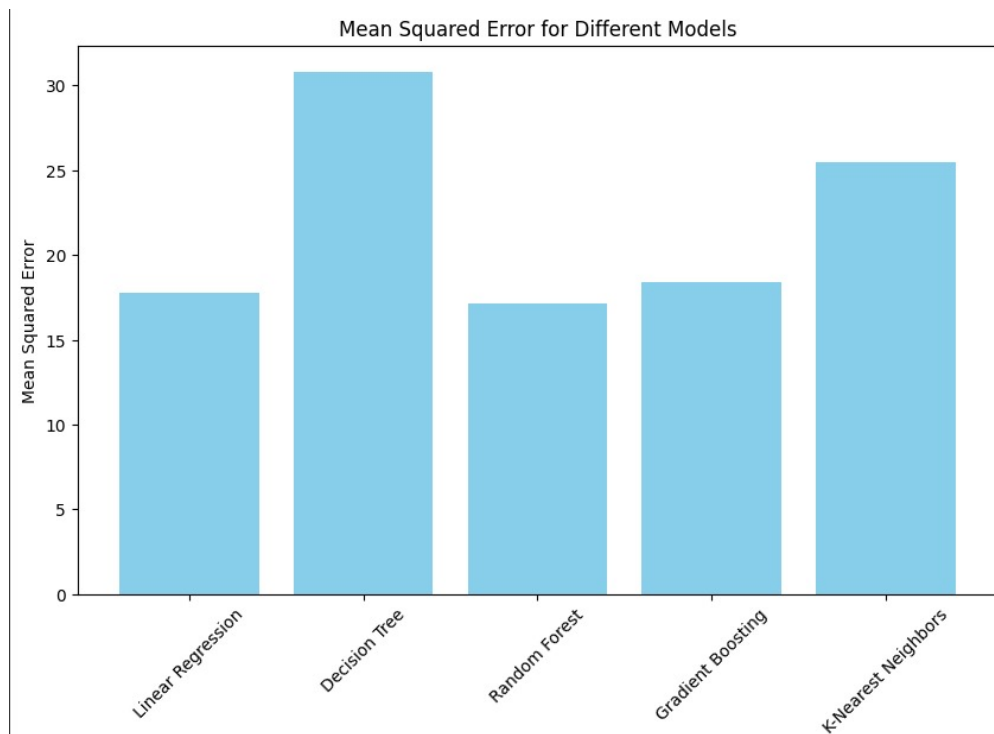


Figura 4.4. Mean Squared Error pentru diferite modele

4.5. Aspecte legate de fiabilitate/ securitate

Datele personale și de sănătate ale utilizatorilor sunt conținute în aplicație, deci securitatea și fiabilitatea acestora trebuie să fie menținute la cele mai înalte niveluri. Pentru a intra în posesia

datelor lor, aplicațiile autorizate pot utiliza Firebase Authentication. Pe lângă asta, datele stocate în Firebase Realtime Database sunt supuse unor reguli stricte de securitate, astfel că numai utilizatorii cu autentificare au dreptul să le acceseze.

Acestea fiind spuse, sunt câteva aspecte și practici pe care le putem lua în calcul pentru a întări fiabilitatea și securitatea pe viitor:

Autentificare multifactor (MFA): implementarea autentificării multifactor pentru a adăuga un nivel suplimentar de securitate. Aceasta poate include utilizarea de SMS-uri, aplicații de autentificare sau autentificare biometrică (amprente, recunoaștere facială).

Token-uri de acces securizate: utilizarea token-urilor de acces securizate și scurte în sesiune pentru a reduce riscul de furt al sesiunii.

Back-up: implementarea unor mecanisme de back-up automat pentru datele utilizatorilor pentru a preveni pierderea informațiilor în caz de defecțiuni hardware sau atacuri cibernetice.

4.6. Rezultate experimentale

Graficul 4.5 reprezintă o histogramă a reziduurilor, care sunt diferențele dintre valorile observate și cele prezise de modelul de regresie. Histograma reziduurilor este importantă pentru a evalua dacă reziduurile urmează o distribuție normală. Dacă histograma are o formă de clopot, similară cu o distribuție normală, acesta este un indicator bun că modelul de regresie liniară este potrivit pentru datele noastre. Reziduurile sunt simetrice în jurul valorii zero, ceea ce sugerează că modelul nu are o părtinire sistematică în supra- sau subestimarea valorilor.

Graficul 4.6, de dispersie al reziduurilor față de valorile prezise este utilizat pentru a evalua dacă există modele neregulate în reziduuri, ceea ce ar putea indica probleme cu modelul. În graficul nostru, punctele nu formează un model clar și par a fi dispersate aleatoriu în jurul liniei roșii (care reprezintă zero). Acesta este un indicator bun, sugerând că modelul de regresie liniară este adecvat.

Graficul leverage vs. reziduuri 4.7 este utilizat pentru a identifica punctele de date care au o influență mare asupra modelului de regresie. Punctele marcate cu numere reprezintă observații care au un leverage mare și pot influența semnificativ modelul de regresie. În cazul nostru, există câteva puncte care se îndepărtează de majoritatea punctelor de date și ar trebui investigate pentru a înțelege de ce au un leverage ridicat. Majoritatea punctelor se află în zona de leverage scăzută, ceea ce indică faptul că modelul nu este influențat în mod excesiv de anumite observații.

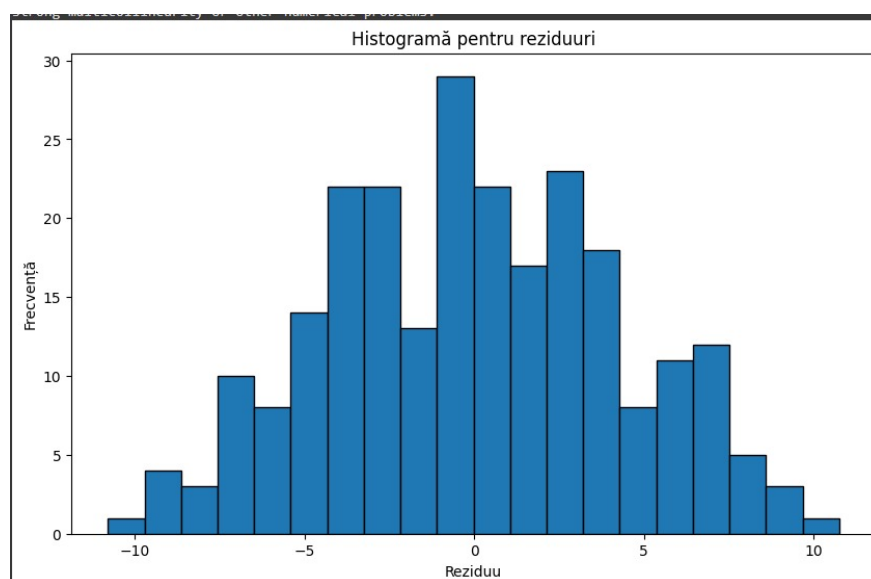


Figura 4.5. Histograma pentru reziduuri

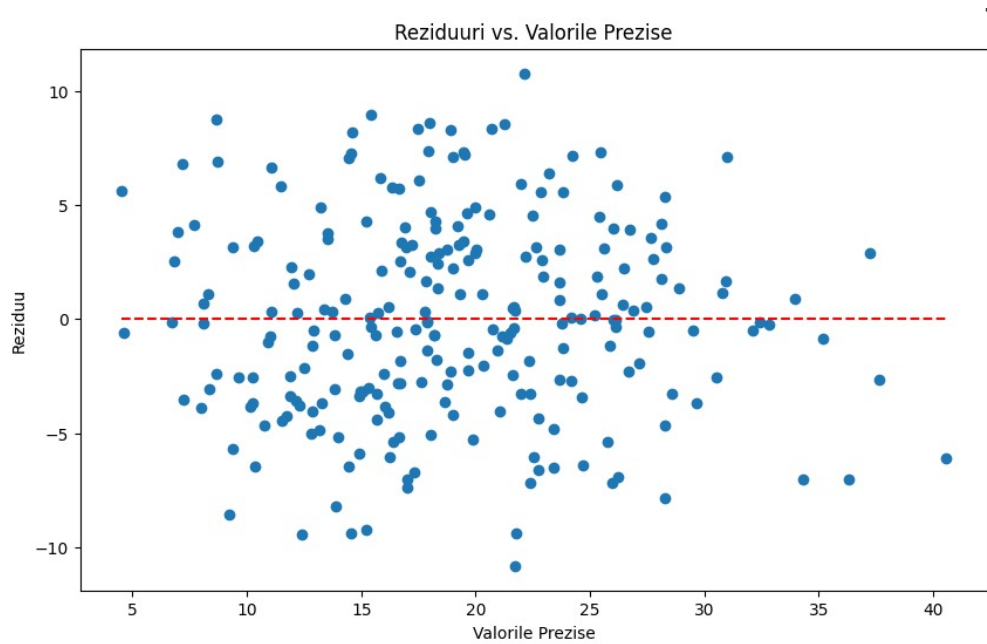


Figura 4.6. Reziduuri vs. valori prezise

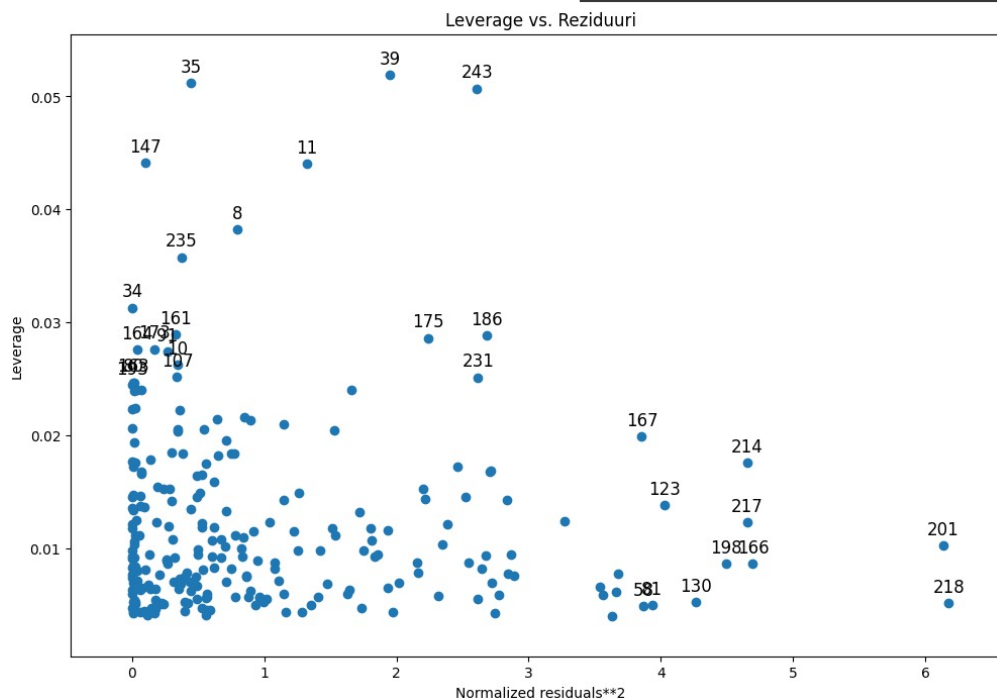


Figura 4.7. Leverage vs. reziduuri

În imaginile 4.8 și 4.9 s-a realizat o comparație între modelul realizat pe învățare automată și un calculator online de grăsime corporală.

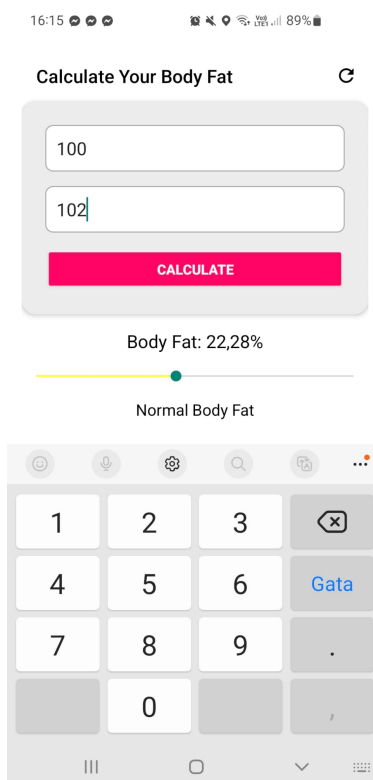


Figura 4.8. Calcularea grăsimii corporale cu metoda aleasă

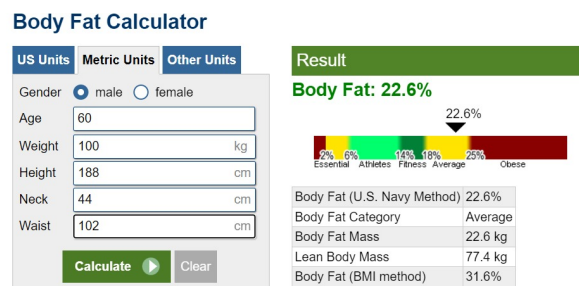


Figura 4.9. Calcularea grăsimii corporale utilizând metoda *US Navy Method* pe un site web [30]

Concluzii

Gradul în care s-a realizat tema propusă

Tema propusă de dezvoltare a unei aplicații mobile pentru estimarea compoziției corporale folosind algoritmi de învățare automată a fost realizată cu succes într-o oarecare măsură, dar încă mai există multe aspecte care pot fi îmbunătățite. Chiar dacă obiectivele inițiale au fost îndeplinite, iar aplicația oferă o modalitate ușor de accesat și precisă pentru monitorizarea compoziției corporale, mai este nevoie de ajustări și optimizări pentru a concura cu soluțiile existente pe piață.

Utilizarea modelului de regresie liniară a adus estimări utile ale compoziției corporale, dar se poate obține o performanță mai bună prin combinarea cu metode avansate și complexe de învățare automată. În plus, interfața cu utilizatorul ar putea fi îmbunătățită pentru a oferi o experiență mai plăcută utilizatorilor, deși este deja funcțională și intuitivă.

În plus, o integrare mai bună a aplicației cu alte servicii și dispozitive de sănătate ar avea un impact valoros. În timp ce aplicația actuală își atinge scopul propus, pentru a reprezenta o soluție competitivă pe piață, trebuie să se continue eforturile de dezvoltare.

4.7. Evidențierea concisă a contribuțiilor/ soluțiilor personale

De-a lungul procesului de dezvoltare a aplicației, s-au implementat soluții personalizate menite să îmbunătățească atât precizia modelului de învățare automată, cât și experiența utilizatorului. Printre acestea se numără:

- Curățarea și preprocesarea datelor: procesarea atentă a datelor a fost un factor esențial pentru succesul modelului de învățare automată. Am eliminat outlierii, am normalizat datele și am folosit metode de imputație pentru a trata valorile lipsă. Am avut nevoie de această etapă pentru a fi siguri că modelul nostru este exact și de încredere.
- Selecția algoritmului de învățare automată: prin selectarea unui model de regresie liniară simplu, dar eficient, s-a reușit să se creeze o soluție care poate funcționa optim pe dispozitive mobile cu resurse reduse. A fost preferat acest model în locul unor algoritmi mai complecși pentru a garanta o implementare rapidă și eficientă.

Interfața prietenoasă utilizatorului: Pentru ca aplicația să fie un succes, a fost esențială realizarea unei interfețe intuitive și atrăgătoare. Pentru a crea o experiență de utilizator fluidă și plăcută, am incorporat principii moderne de design UI/UX. Utilizatorii pot introduce datele rapid și accesa cu ușurință funcționalitățile principale prin intermediul interfeței aplicației.

- Monitorizarea pașilor și calculul necesarului caloric: adăugarea unui sistem de monitorizare a pașilor și calculul necesarului caloric zilnic prin utilizarea senzorilor telefonului a adus o valoare semnificativă aplicației. Utilizatorii beneficiază de informații relevante pentru monitorizarea activității fizice și gestionarea greutății prin intermediul acestor funcționalități.
- Documentare și transparență: m-am asigurat că am documentat cu atenție fiecare etapă a proiectului. Aceasta nu doar că a ușurat dezvoltarea, ci și a garantat transparența procesului, oferind posibilitatea altor programatori să înțeleagă și să reproducă rezultatele.

Comparație cu alte proiecte similare

Comparată cu alte aplicații existente, soluția propusă se remarcă prin costuri mai accesibile și absența necesității unui echipament suplimentar. Un caz concret este reprezentat de Fitbit Aria, care implică achiziționarea unui dispozitiv separat. Aceasta poate implica costuri semnificative pentru utilizatori.

În schimb, aplicația noastră folosește doar smartphone-ul utilizatorului, eliminând nevoia de hardware suplimentar. Integrarea cu Firebase oferă utilizatorilor o experiență fluidă prin sincronizarea datelor în timp real și accesibilitatea de pe orice dispozitiv.

Aplicația care a fost dezvoltată în cadrul acestui proiect se remarcă prin estimarea compoziției corporale folosind algoritmi de învățare automată, în timp ce aplicații precum Apple Health și Google Fit oferă caracteristici similare. Așa cum face aplicația noastră, Apple Health și Google Fit oferă o gamă largă de funcționalități de sănătate și fitness, dar nu se concentrează în mod specific pe estimarea compoziției corporale.

Posibile direcții de dezvoltare

Pe viitor, aplicația poate fi îmbunătățită prin:

- Integrarea unor funcționalități suplimentare, cum ar fi monitorizarea nutriției și a nivelului de hidratare. Aceste funcționalități ar permite utilizatorilor să înregistreze și să monitorizeze consumul zilnic de alimente și apă.
- Extinderea compatibilității aplicației cu dispozitive wearable pentru a colecta date biometrice suplimentare în timp real. Dispozitivele wearable, cum ar fi brățelele de fitness și ceasurile inteligente, pot furniza date detaliate despre activitatea fizică, ritmul cardiac și alți parametri de sănătate, îmbunătățind astfel capacitatea aplicației de a monitoriza starea de sănătate a utilizatorilor.
- Implementarea unor funcții de coaching personalizat, bazate pe datele colectate, pentru a oferi utilizatorilor recomandări specifice privind sănătatea și fitness-ul. Funcțiile de coaching personalizat ar putea include sugestii de exerciții fizice, planuri de dietă și obiective de sănătate adaptate nevoilor și preferințelor individuale ale utilizatorilor.
- Dezvoltarea unei comunități online în cadrul aplicației unde utilizatorii pot împărtăși experiențe, sfaturi și motivație, contribuind astfel la creșterea angajamentului.
- Implementarea unor funcții cum ar fi recompensele și realizările, pentru a încuraja utilizatorii să-și atingă obiectivele de sănătate și fitness într-un mod distractiv și motivant.
- Extinderea suportului pentru diverse limbi și culturi pentru a face aplicația accesibilă unui public global.

Lecții învățate pe parcursul dezvoltării lucrării de licență

Dezvoltarea acestei aplicații a oferit numeroase lecții valoroase atât din punct de vedere tehnic, cât și personal. Pe lângă aspectele specifice aplicației, am învățat:

- Dezvoltarea unui limbaj academic: pe parcursul cercetării și scrierii lucrării am învățat să dezvolt un limbaj academic coerent și clar, esențial pentru a comunica ideile și rezultatele într-un mod profesionist. Acest proces a implicat analizarea unor articole, cărți, lucrări științifice, înțelegerea structurii și stilului academic și aplicarea acestor cunoștințe în scrierea propriei lucrări.
- Extracția informațiilor importante: un alt aspect important a fost capacitatea de a extrage informații relevante dintr-o varietate de surse. Am învățat să identific sursele credibile, să analizez și să interpretez datele și să le integrez în mod eficient în contextul lucrării mele.
- Gestionarea proiectului: dezvoltarea unei aplicații funcționale a necesitat o bună gestionare a timpului și resurselor. Am învățat să prioritizez sarcinile, să stabilesc termene realiste și să monitorizez progresul pentru a asigura finalizarea la timp a proiectului.

- Colaborarea și feedback-ul: interacțiunea cu colegii, profesorii și mentorii a fost esențială pentru îmbunătățirea constantă a proiectului. Feedback-ul primit a contribuit la identificarea punctelor slabe și la implementarea unor soluții mai eficiente și mai inovatoare.
- Adaptabilitatea: pe parcursul dezvoltării aplicației, am întâlnit numeroase provocări și obstacole tehnice. Am învățat să fiu flexibil și să mă adaptez rapid la schimbări, găsind soluții alternative.

Bibliografie

- [1] J. Higgins, "Smartphone applications for patients' health & fitness," *The American Journal of Medicine*, vol. 129, pp. 11–19, 2015, DOI: [10.1016/j.amjmed.2015.05.038](https://doi.org/10.1016/j.amjmed.2015.05.038).
- [2] "mhealth apps," <https://www.statista.com/statistics/779919/health-apps-available-google-play-worldwide/>, Ultima accesare: 28.06.2024.
- [3] "Fitbit Aria," <https://www.fitbit.com/global/us/products/scales/aria-air>, Ultima accesare: 20.06.2024.
- [4] "MyFitnessCompanion," <https://play.google.com/store/apps/details?id=com.fitness.myfitnesscompanion&hl=ro&gl=US>, Ultima accesare: 20.06.2024.
- [5] "Apple health," <https://www.apple.com/ios/health/>, Ultima accesare: 20.06.2024.
- [6] "Google fit," <https://www.google.com/fit/>, Ultima accesare: 20.06.2024.
- [7] "Samsung health," <https://www.samsung.com/global/galaxy/apps/samsung-health/>, Ultima accesare: 20.06.2024.
- [8] "Strava," <https://www.strava.com/>, Ultima accesare: 20.06.2024.
- [9] A. Balapour, I. Reychav, R. Sabherwal, and J. Azuri, "Mobile technology identity and self-efficacy: Implications for the adoption of clinically supported mobile health apps," *International Journal of Information Management*, vol. 49, pp. 58–68, 2019, DOI: [10.1016/j.ijinfomgt.2019.03.005](https://doi.org/10.1016/j.ijinfomgt.2019.03.005).
- [10] S. Brown, "Exploring benefits and ethical challenges in the rise of mhealth (mobile health-care) technology for the common," *Journal of Medical Ethics*, 2019.
- [11] "Android Studio documentation," <https://developer.android.com/studio/intro/>, Ultima accesare: 10.06.2024.
- [12] "Firebase for android documentation," <https://firebase.google.com/docs/android/setup>, Ultima accesare: 28.06.2024.
- [13] "Scikit-learn linear models," https://scikit-learn.org/stable/modules/linear_model.html, Ultima accesare: 20.06.2024.
- [14] R. Mahesh, G. Chowdary, and D. Abhiram, "Role of artificial intelligence (ai) in the health care and its impact & applications," *Technix International Journal for Engineering Research*, vol. 10, no. 8, pp. b60–b67, 2023.
- [15] J. Calvillo-Arbizu, I. Roman-Martinez, and J. Reina-Tosina, "Internet of things in health: Requirements, issues, and gaps," *Computer Methods and Programs in Medicine*, vol. 208, p. 106231, 2021, DOI: [10.1016/j.cmpb.2021.106231](https://doi.org/10.1016/j.cmpb.2021.106231).
- [16] "Realitatea virtuală: Ce este și unde o putem folosi," <https://maraton.digital/realitatea-virtuala-ce-este-si-unde-o-putem-folosi/>, Ultima accesare: 30.06.2024.
- [17] R. Angel, "Cum revolutioneaza realitatea virtuala antrenamentele," <https://fitness-nation.ro/blogs/articole/cum-revolutioneaza-realitatea-virtual-antrenamentele.html>, 2024, Ultima accesare: 28.06.2024.

- [18] A. Abdullah, A. Ismael, A. Rashid, A. Abou-ElNour, and M. Tarique, “Real time wireless health monitoring application using mobile devices,” *International Journal of Computer Networks & Communications*, vol. 7, pp. 13–30, 2015, DOI: [10.5121/ijcnc.2015.7302](https://doi.org/10.5121/ijcnc.2015.7302).
- [19] R. N. Madeira, H. Germano, P. Macedo, and N. Correia, “Personalising the user experience of a mobile health application towards patient engagement,” *Procedia Computer Science*, vol. 141, pp. 428–433, 2018, DOI: [10.1016/j.procs.2018.10.173](https://doi.org/10.1016/j.procs.2018.10.173).
- [20] “Body fat prediction dataset,” <https://www.kaggle.com/datasets/fedesoriano/body-fat-prediction-dataset/>, Ultima accesare: 27.06.2024.
- [21] A. Behnke and J. Wilmore, *Evaluation and Regulation of Body Build and Composition*, ser. International research monograph series in physical education. Prentice-Hall, 1974. [Online]. Available: <https://books.google.ro/books?id=8MpqAAAAMAAJ>
- [22] K. Penrose, A. Nelson, and A. Fisher, “Facsm. generalized body composition prediction equation for men using simple measurement techniques,” *Medicine & Science in Sports & Exercise*, vol. 17, no. 2, 1985, DOI: [http://dx.doi.org/10.1249/00005768-198504000-00037](https://doi.org/10.1249/00005768-198504000-00037).
- [23] W. Siri, “The gross composition of the body,” *Adv Biol Med Phys*, vol. 4, pp. 239–280, 1956, DOI: [10.1016/b978-1-4832-3110-5.50011-x](https://doi.org/10.1016/b978-1-4832-3110-5.50011-x).
- [24] W. Sheikh and N. Sheikh, “A model-view-viewmodel (mvvm) application framework for hearing impairment diagnosis - type dependency architecture,” in *Proceedings of the 2020 Intermountain Engineering, Technology and Computing (IETC)*, Orem, UT, USA, 2020, pp. 1–6, DOI: [10.1109/IETC47856.2020.9249181](https://doi.org/10.1109/IETC47856.2020.9249181).
- [25] “Model–view–viewmodel,” <https://en.wikipedia.org/wiki/Model%E2%80%93view%E2%80%93viewmodel>, Ultima accesare: 30.06.2024.
- [26] “Stat 628 - module 2,” <https://github.com/yzeng58/bodyfat-calculator/blob/master/Jupyter%20notebook%20summary.ipynb>, Ultima accesare: 30.06.2024.
- [27] Z. Fan, R. Chiong, Z. Hu, F. Keivani, and F. Chiong, “Body fat prediction through feature extraction based on anthropometric and laboratory measurements,” *PLoS One*, vol. 17, no. 2, p. e0263333, 2022, DOI: [10.1371/journal.pone.0263333](https://doi.org/10.1371/journal.pone.0263333).
- [28] N. Mahesh, P. B. Pati, K. Deepa, and S. Yanan, “Body fat prediction using various regression techniques,” in *2023 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI)*, 2023, pp. 1–5.
- [29] D. DL, S. RJ, C. SA, L. M, C. AC, S. RM, and C. C. W, “Body composition methods: comparisons and interpretation,” *J Diabetes Sci Technol*, vol. 2, no. 6, pp. 1139–1146, 2008, DOI: [10.1177/193229680800200623](https://doi.org/10.1177/193229680800200623).
- [30] “Calculator.net,” <https://www.calculator.net/body-fat-calculator.html>, Ultima accesare: 30.06.2024.