

Android aplikacija za planiranje zdrave prehrane

Kovačević, Matej

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:200:549081>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-22**

Repository / Repozitorij:

[Faculty of Electrical Engineering, Computer Science and Information Technology Osijek](#)



**SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU
FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I
INFORMACIJSKIH TEHNOLOGIJA**

Sveučilišni studij

**ANDROID APLIKACIJA ZA PLANIRANJE ZDRAVE
PREHRANE**

Završni rad

Matej Kovačević

Osijek, 2020.

SADRŽAJ

1.	Uvod.....	1
1.1.	Zadatak završnog rada.....	2
2.	Zdrava prehrana i trendovi	3
2.1.	Prehrana i kronične nezarazne bolesti	3
2.2.	Smjernice za zdravu prehranu	3
3.	Prehrambena piramida i tanjur zdrave prehrane	5
4.	Potreba pojedinca za energijom	9
4.1.	Izračunavanje indeksa tjelesne mase pojedinca	9
4.2.	Izračunavanje ukupnih energetske potrebe pojedinca	10
5.	Postojeća rješenja planiranja zdrave prehrane	13
5.1.	Usporedba s postojećim rješenjima.....	13
5.1.1.	<i>Eat this much</i>	13
5.1.2.	<i>MyNetDiary</i>	14
5.1.3.	<i>MyHealthPyramid</i>	14
5.1.4.	Usporedba s prethodno nabrojanim aplikacijama	15
5.2.	Koncept aplikacije <i>StayHealthy</i>	15
6.	Korištena tehnologija	17
6.1.	Razvojna okolina Androida.....	17
6.1.1.	Općenito o operacijskom sustavu Androida.....	17
6.1.2.	Arhitektura Androida	18
6.2.	Android Studio	19
6.2.1.	Android SDK.....	20
6.3.	Java programski jezik.....	20
6.4.	XML.....	21
6.5.	Firestore	22
6.6.	GIMP.....	22
7.	Razvoj aplikacije <i>StayHealthy</i>	24
7.1.	Gradle	24
7.2.	AndroidManifest datoteka.....	25
7.3.	Aktivnost <i>MainActivity</i>	25
7.4.	Aktivnost <i>SignUp</i>	26
7.5.	Aktivnost <i>ParameterActivity</i>	27

7.6. Aktivnost <i>HomeActivity</i>	29
7.7. Aktivnost <i>FoodMenuActivity</i>	34
8. Zaključak.....	38
Literatura	39
Sažetak	42
Abstract	43
Životopis.....	44
Popis slika	45
Popis tablica	46
Prilozi	47

1. UVOD

Prekomjerni unos energije, loše prehrambene navike te smanjena tjelesna aktivnost i sjedilački način života tipičan za moderno društvo, glavni su razlozi razvoja kroničnih nezaraznih bolesti (prekomjerne težine i pretilosti, kardiovaskularnih bolesti, raka, hipertenzije, dijabetesa tipa 2 i osteoporoze) i povećane smrtnosti stanovništva. Organizacija za hranu i poljoprivredu UN-a (*FAO*) i Svjetska zdravstvena organizacija (*WHO*) pozvale su države da u svrhu „promocije odgovarajuće prehrane i zdravog načina života“ donesu za svoje stanovništvo prehrambene smjernice (engl. *Food-based dietary guidelines, FBDG*). Jedan od najčešće korištenih *FBDG*-ova koji je postao i sinonim za optimiranje obroka u sustavu zdrave prehrane je „prehrambena piramida“ i njezina zadnja američka verzija iz 2011. godine pod nazivom *MyPlate*. Kako bi se osigurao personalizirani pristup prehrani koji je temeljen na životnom stilu, dobi, spolu i fizičkim karakteristikama korisnika, danas se sve češće koriste različite mobilne i računalne aplikacije. U ovom radu razvijena je aplikacija koja omogućava personalizirano optimiranje prehrane s obzirom na kalorijski unos hrane te s obzirom na zastupljenost pojedinih skupina hrane u skladu s prehrambenom piramidom, odnosno s *MyPlate*.

U drugom poglavlju opisan je pojam zdrave prehrane i utjecaj prehrane na zdravlje pojedinca te su objašnjeni principi po kojima se određuju prehrambene smjernice kao znanstveno utemeljene preporuke za zdravu prehranu određenog područja ili dobne skupine. U trećem poglavlju prikazana je povijest nastanka prehrambene piramide kao sustava optimiranja prehrane, odnosno obroka u kojima je zastupljena pojedina skupina hrane s posebnim naglaskom na najnoviju verziju *MyPlate* iz 2011. godine. Četvrto poglavlje objašnjava formule i izraze vezane za indeks tjelesne mase i energetske potrebe pojedinca, a koji su implementirani u aplikaciju kako bi se vršili potrebni izračuni za planiranje prehrane korisnika. Pregled postojećih sličnih programskih rješenja i usporedba s aplikacijom opisanom u ovom radu nalazi se u petom poglavlju, a isto tako opisan je i način rada aplikacije. U šestom poglavlju dana je teorijska podloga aplikacije, odnosno opisana je Android razvojna okolina, programski jezici i sve ostale tehnologije koje su bile potrebne u kreiranju aplikacije. Posljednje poglavlje pruža uvid u najbitnije sastavnice aplikacije, dijelove programskog koda i izgled korisničkog zaslona. Kako bi se programski kod i opisi pojedinih zaslona mogli lakše pratiti, posljednje poglavlje sadrži i slike konkretnih dijelova.

U ovom radu razvijena je aplikacija *StayHealthy* u razvojnom okruženju Android Studio što podrazumijeva poznavanje njegovih osnovnih alata i programskog jezika Java kojim je aplikacija napisana. Za pohranu podataka korištena je baza podataka jedne od trenutno najpopularnijih platformi Firebase, a za potrebe uređivanja slika korištenih u aplikaciji primijenjen je uređivač slika GIMP.

1.1. Zadatak završnog rada

Opisati pojam i problematiku zdrave prehrane. Predložiti funkcionalnosti koje bi trebala imati Android aplikacija koja bi se bavila planiranjem zdrave prehrane. Izraditi Android aplikaciju pomoću koje korisnik može planirati zdravu prehranu s ciljem normaliziranja vrijednosti indeksa tjelesne mase. Podatke o stanju organizma, obrocima i nutrijentima pohraniti u odgovarajuću bazu podataka.

2. ZDRAVA PREHRANA I TRENDОВI

Pravilna prehrana nužna je za pravilan razvitak i rad tjelesnih organa, za rast, razmnožavanje i održanje, za najučinkovitiju tjelesnu i umnu aktivnost, za očuvanje imuniteta i sposobnosti zacjeljivanja. Premda su do danas brojni kemijski sastojci hrane istraženi i poznata je njihova uloga i važnost za pravilan rad ljudskih organa, treba naglasiti da na učinkovitost svakoga sastojka, više ili manje, utječu svi drugi sastojci u hrani. Zbog toga treba, tijekom cijeloga života, omogućiti primjeren dnevni unos svih potrebnih hranjivih sastojaka hrane, jer preveliki unos hrane, neuravnotežena prehrana ili neispravna hrana i voda mogu dovesti do bolesti. Razvitak prehrambene tehnologije i suvremeni način života utjecali su na promjenu prehrambenih navika pojedinaca, društvenih skupina i cijeloga društva.

2.1. Prehrana i kronične nezarazne bolesti

Loša prehrana i sjedilački način života glavni su razlozi razvoja kroničnih nezaraznih bolesti (prekomjerne težine i pretilosti, kardiovaskularnih bolesti, raka, hipertenzije, dijabetesa tipa 2 i osteoporoze) i povećane smrtnosti stanovništva. Procjenjuje se da više od polovice europskog odraslog stanovništva ima prekomjernu tjelesnu masu, a 15,9% je pretilo [1]. Navedene bolesti veliki su javnozdravstveni problem modernih društava sa znatnim ekonomskim troškovima (liječenje, bolovanja i dr.) koje se procjenjuju na 0,7 – 2,8% BDP-a godišnje [2].

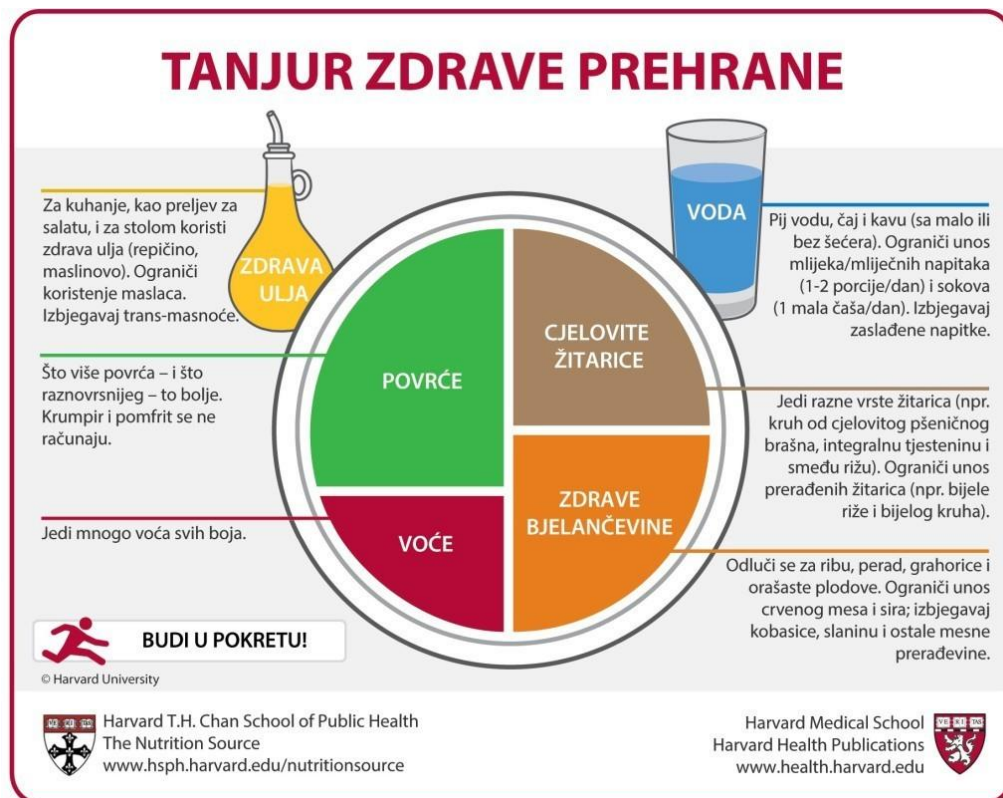
2.2. Smjernice za zdravu prehranu

Na međunarodnoj konferenciji o prehrani 1992. koju su organizirali Organizacija za prehranu i poljoprivredu Ujedinjenih naroda (engl. *United Nations Food and Agricultural Organization – FAO*) i Svjetska zdravstvena organizacija (engl. *World Health Organization – WHO*) utvrđen je plan djelovanja koji uključuje „promociju odgovarajuće prehrane i zdravog načina života“ i poziva svjetske vlade da u javnosti promoviraju kvalitativne i kvantitativne prehrambene smjernice važne za različite dobne skupine i životne stilove koji odgovaraju stanovništvu pojedine zemlje, odnosno *Food-based dietary guidelines* (FBDG). Prehrambene smjernice su definirane kao znanstveno utemeljene preporuke u obliku smjernica za zdravu prehranu. Oni su lako razumljivi dokumenti namijenjeni potrošačima kao i kreatorima politika, zdravstvenim radnicima, prehrambenoj industriji te se koriste i u obrazovne svrhe. U prošlosti su prevencija bolesti i preporuke o hranjivim tvarima dominirale u procesu uspostavljanja prehrambenih smjernica. Međutim, znanstveni napredak i društveni razvoj kao što je promjena životnog stila, interes za personalizirano zdravlje i zabrinutost za održivost razvoja i zaštitu okoliša, zahtijevaju preusmjeravanje prehrambenih

smjernica kako bi obuhvatio širi spektar prehrambenog ponašanja. Europska agencija za sigurnost hrane (engl. *European Food Safety Authority* - EFSA) izdala je znanstveno mišljenje o tome kako uspostaviti FBDG u Europi [3]. EFSA se usredotočila na proces razvoja FBDG-a i predložila postupni pristup u kojem je identifikacija odnosa prehrane i zdravlja središnja polazna točka te da se FBDG-ovi trebaju uspostaviti posebno za svaku zemlju ili regiju zbog velikih razlika u prehranbenim/kulturnim navikama i dostupnosti prehranbenih proizvoda [3]. Naime, proizvodnja hrane, distribucija, troškovi, sklonosti, navike, kulinarske prakse, gastronomija općenito razlikuju se značajno od zemlje do zemlje, pa čak i između pojedinih regija za što je najbolji primjer Hrvatska (kontinentalna, planinska i primorska). Jedan od najčešće korištenih FBDG-ova koji je postao i sinonim za optimiranje obroka u sustavu zdrave prehrane su tzv. prehranbene piramide.

3. PREHRAMBENA PIRAMIDA I TANJUR ZDRAVE PREHRANE

Prehrambena piramida predstavlja sustav optimiranja prehrane, odnosno obroka u kojima je zastupljena pojedina skupina hrane. Prva prehrambena piramida objavljena je u Švedskoj 1974., druga 1992. u SAD pod nazivom „*Food Guide Pyramid*“ ili „*Eating Right Pyramid*“, treća verzija (nadopunjena), također u SAD pod nazivom „*MyPyramid*“ te zadnja verzija 2011. godine pod nazivom „*MyPlate*“ prikazana na slici 3.1. Piramida zdrave prehrane (engl. *The Healthy eating pyramid*), prikazana na slici 3.2., koju je objavila Škola javnog zdravlja harvardskog sveučilišta također se bavi drugim aspektima zdravog načina života - vježbanjem, kontrolom tjelesne težine, dodacima vitamina D i multivitamina te umjerenošću alkohola za osobe koje piju.



Slika 3.1. Tanjur zdrave hrane "MyPlate", izvor: [4]

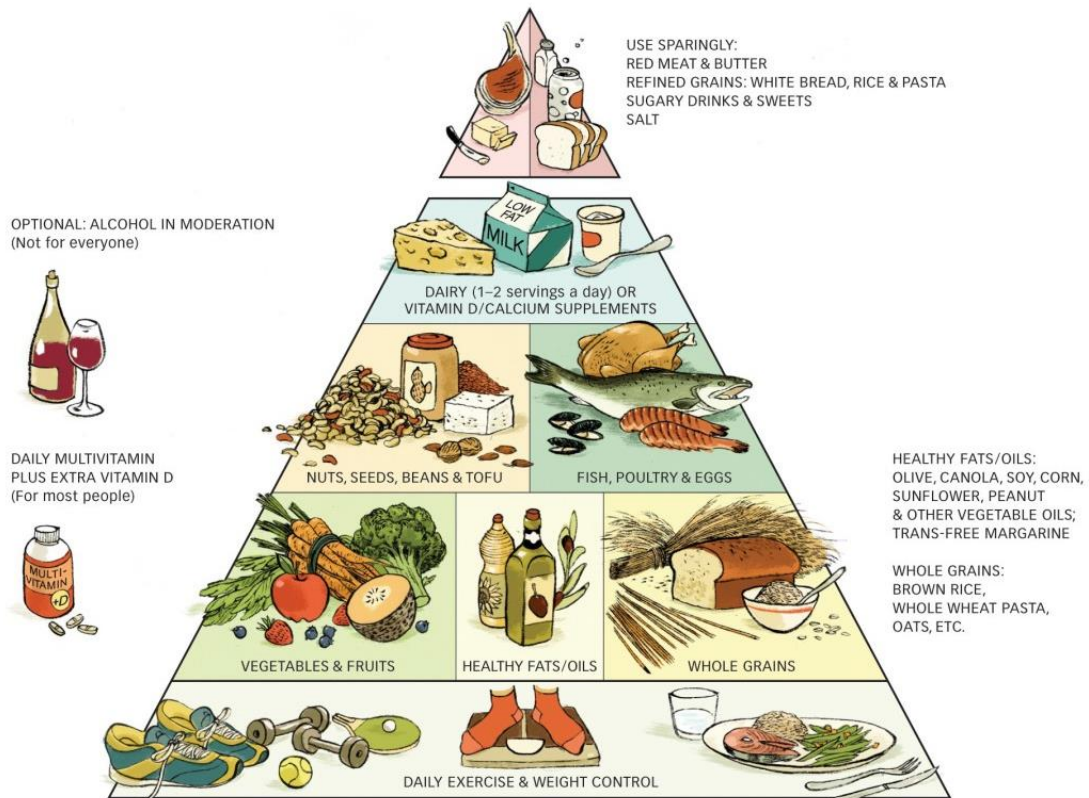
Prehrambenu piramidu (engl. *Food Pyramid*) 2002. godine objavili su u suradnji i WHO i FAO pri čemu je struktura piramide u nekim aspektima slična prehrambenoj piramidi SAD-a, ali postoje jasne razlike između vrsta masti i veća se pozornost pridaje razlikama u kojoj se ugljikohidrati kategoriziraju na osnovi slobodnih šećera u odnosu na šećere u njihovom prirodnom obliku. Također, uz prehrambenu piramidu dane su i preporuke za pojedine sastojke hrane, što je vidljivo u tablici 3.1.

Tablica 3.1. Prehrambene preporuke WHO i FAO, izvor: [5]

Sastojak hrane	Preporuke WHO iz 1989.	Preporuke WHO/FAO iz 2002. godine
ukupne masti	15 - 30%	15 - 30%
zasićene masne kiseline (SFAs)	0 - 10%	<10%
polinezasićene masne kiseline (PUFAs)	3 - 7%	6 - 10%
n-6 PUFAs	-	5 - 8%
n-3 PUFAs	-	1 - 2%
trans-masne kiseline	-	<1%
mononezasićene masne kiseline (MUFAs)	-	različito
Ukupni šećeri	55 - 75%	55 - 75%
Slobodni šećeri	0 - 10%	<10%
složeni ugljikohidrati	50 - 70%	ne preporučuje se
proteini	10 - 15%	10 - 15%
kolesterol	0 – 300 mg/dan	< 300 mg/dan
Natrijev klorid (Na)	< 6 g/dan	< 5 g/dan (< 2 g/dan)
voće i povrće	≥ 400 g/dan	≥ 400 g/dan
mahunarke, orašasto voće, sjemenke	≥ 30 g/dan (kao dio 400 g voća i povrća)	-
ukupna prehrambena vlakna	27 - 40 g/dan	iz hrane
ne-škrobni polisaharidi (NSP)	16 - 24 g/dan	iz hrane

MyPlate je podijeljen u četiri odjeljka s oko 30 % žitarica, 40 % povrća, 10 % voća i 20 % proteina, a prati ih manji krug koji predstavlja mliječne proizvode, poput čaše mlijeka ili jogurta sa smanjenim udjelom mliječne masti. Tanjur pravilne prehrane predstavlja vodič za planiranje dobro balansiranih obroka te služi kao alat za održavanje pravilne prehrane i zdravog načina života. On neće definirati broj kalorija koje svaka osoba treba svakodnevno unijeti. Tanjur zdrave prehrane poručuje da prije svega treba obratiti pažnju na kvalitetu prehrane a uz to voditi računa o umjerenosti, raznolikosti i uravnoteženosti prehrane. Prema *MyPlate* preporuka je da se većina obroka, odnosno ½ ukupnog dnevnog unosa grane sastoji od voća i povrća raznih boja i vrsta,

izuzev krumpira zbog negativnog djelovanja na razinu šećera u krvi. Preporučuje se da cjelovite žitarice (integralna pšenica, ječam, kvinoja, zob, smeđa riža i obroci pripremljeni na bazi ovih žitarica, npr. integralna tjestenina), koje imaju manje negativan utjecaj na razinu šećera i inzulina u krvi nego bijeli kruh, bijela riža i druge prerađene žitarice čine $\frac{1}{4}$ dnevnog unosa hrane. Uz voće, povrće i cjelovite žitarice, veliki udio, odnosno $\frac{1}{4}$ dnevnog unosa hrane trebaju činiti riba, piletina, grahorice i orašasti plodovi koji su zdravi i raznoliki izvori bjelančevina. Unos crvenog mesa treba ograničiti, a u potpunosti izbjegavati mesne prerađevine, kao što su kobasice i slanina.



Slika 3.2. Prehrambena piramida prema Školi javnog zdravlja T. H. Chan harvardskog sveučilišta, izvor: [4]

Od ulja treba koristiti zdrava biljna ulja kao što su maslinovo, repičino, sojino, kukuruzno, suncokretovo, kikirikijevo i druga, te u potpunosti izbjegavati djelomično hidrogenizirane masnoće, koje sadrže nezdrave trans-masnoće. Poželjno je izbjegavati zaslađene napitke te ograničiti unos mlijeka i mliječnih proizvoda na jednu do dvije porcije dnevno. Crvena silueta u pokretu koja upotpunjuje *MyPlate* služi kao podsjetnik da je i fizička aktivnost važna za kontrolu tjelesne težine. Prema piramidi prehrane, a na osnovi nekih zajedničkih nutritivnih karakteristika, namirnice se najčešće dijele u šest skupina: žitarice i proizvodi od žita, voće i povrće, mlijeko i mliječni proizvodi, meso, riba i jaja, masti i ulja, dodaci hrani (sol, začini, šećer, umjetni

zaslađivači). Odnos između površine koju razina zauzima u piramidi i zastupljenosti pojedine skupine namirnica u prehrani je proporcionalna. Znači, razine bliže bazi imaju veću površinu, pa se i namirnice iz tih skupina trebaju konzumirati u većim količinama i suprotno, više razine imaju manju površinu, pa bi i unos tih namirnica trebao biti manji. Osnovna je poruka piramide pravilne prehrane jednostavna: raznolika je prehrana nužna za zdravlje stoga je svaki dan potrebno u adekvatnoj količini birati namirnice iz svih tih 6 skupina.

4. POTREBA POJEDINCA ZA ENERGIJOM

Energetska vrijednost hrane određena je količinom topline, koju oslobađaju njezini sastavni dijelovi pri potpunom sagorijevanju u kalorimetru. Prema međunarodnomu sustavu (SI) jedinica za mjerenje energije je jedan J (džul, engl. joule), ali se vrlo često energetska vrijednost hrane ili potreba za energijom izražavaju u kalorijama (cal) ili kilokalorijama (kcal) ($1 \text{ kcal} = 4,184 \text{ kJ}$ ili $1 \text{ kJ} = 0,239 \text{ kcal}$). Glavni izvor energije koji se u tijelu koriste za rad i održavanje tjelesne topline su ugljikohidrati, masti i bjelančevine. Potrošnja alkohola također znatno pridonosi ukupnomu unosu energije. Prosječna količina energije oslobođena (sagorijevanjem) po jedinici mase tvari izračunava se na sljedeći način: bjelančevine 4 kcal/g (17 kJ/g), masti 9 kcal/g (37 kJ/g), ugljikohidrati 4 kcal/g (17 kJ/g), alkohol 7 kcal/g (29 kJ/g), odnosno, 1 g bjelančevina \approx 1 g ugljikohidrata \approx 0,44 g masti. Pravilna prehrana mora zadovoljiti ukupne dnevne potrebe tijela za energijom [6].

Zdrava energetska ravnoteža u organizmu čovjeka postiže se kada je energetske unos (zbroy energije dobivene iz hrane i pića) izjednačen s energetske gubitcima uslijed potrošnje energije na bazalni metabolizam, termički učinak hrane i dobrovoljnu tjelesnu aktivnost. Prosječan dnevni unos kalorija za žene iznosi 2.000 do 2.100 kcal, a za muškarce 2.700 - 2.900 kcal [6]. Osobna potreba za hranjivim tvarima zavisi o mnogim čimbenicima: životnoj dobi, spolu, visini i tjelesnoj masi, količini tjelesnoga rada te klimatskim uvjetima u kojima osoba živi itd. Održavanje ravnoteže unosa energije hranom i potrošnje energije može biti zahtjevno kod veoma velikih tjelesnih napora (na primjer teške sportske vježbe, teški tjelesni rad itd.) ili kada je mišićni rad veoma mali. Tijelo zadržava stalnu tjelesnu masu kada je unos energije jednak potrošku energije. U slučaju prekomjernog unosa energije povećava se i tjelesna masa. U prosjeku, za svako povećanje ili smanjenje od oko 0,5 kilograma tjelesne mase, treba unijeti ili potrošiti dodatnih 3500 kcal [6].

4.1. Izračunavanje indeksa tjelesne mase pojedinca

Uobičajeni način za ocjenu prehrambenoga stanja odraslih osoba je izračunavanje indeksa tjelesne mase (ITM). Indeks tjelesne mase računa se tako da se masa u kilogramima podijeli s kvadratom visine u metrima [6]:

$$ITM = \frac{\text{tjelesna masa (kg)}}{(\text{visina u metrima})^2} \quad (4 - 1)$$

Prema mjerilima Svjetske zdravstvene organizacije za ocjenu prehrambenoga stanja prema vrijednosti ITM odraslih osoba (osim trudnica) [6]:

- ITM < 18,5 = neuhranjen,
- 18,5 - 24,99 = poželjna tjelesna masa,
- 25,0 - 29,99 = prekomjerna tjelesna masa,
- 30,00 - 34,90 = I. stupanj debljine,
- 35,00 - 39,99 = II. stupanj debljine,
- ≥ 40 = III. stupanj debljine.

ITM je dobar pokazatelj debljine, ali postoje i situacije u kojima ITM pokazuje lažne vrijednosti, npr. veliki ITM u slučaju mišićavog tijela ili pojačanog nakupljanja tekućine u tijelu. Često se debljina utvrđuje procjenom nakupljanja masnog tkiva u području trbuha, odnosno mjerenjem opsega struka u visini pupka (vrijednost > 94 cm za muškarce i > 80 cm za žene znači da je osoba debela). [6]

4.2. Izračunavanje ukupnih energetske potreba pojedinca

Postoje više ili manje složene jednadžbe za izračunavanje EBM pojedinca. Međutim, ne može se reći da bilo koja od tih jednadžbi omogućuje izračun energije EBM 100% točno. Na ukupnu tjelesnu potrebu za energijom utječe [6]:

a) **Energija bazalnoga metabolizma - EBM** (engl. *Basal Metabolic Rate* - BMR)

Od ukupne količine energije koju čovjek potroši tijekom dana (24 sata) najviše otpada na potrebe bazalnoga metabolizma (BM). Brzina bazalnog metabolizma (BBM) je količina energije koja je potrebna za održavanje normalne metaboličke aktivnosti tijela, kao što su disanje, održavanje tjelesne temperature i probava. Točnije, to je količina energije potrebna u mirovanju ili spavanju bez dodatnih aktivnosti. Potrošena energija dovoljna je samo za rad vitalnih organa poput srca, pluća, živčanog sustava, bubrega, jetre, crijeva, spolnih organa, mišića i kože. Prema istraživanjima brzina metabolizma za vrijeme spavanja je 90 -100% od BBM te osobe.

b) **Energija potrebna za rad – ER**

Od 100 do 3.000 kcal ovisno o intenzitetu tjelesne aktivnosti.

c) **Energija potrebna za probavljanja hrane – EPH**

To je energija potrebna za probavu, upijanje, prijenos, pohranu i metabolizam hranjivih tvari. Količina energije koja je potrebna za probavljanje hrane zavisi o omjeru

makrohranjivih tvari u obroku, jer se za metaboličku oksidaciju ili pohranu masti troši 2 - 3% od unijete energije, za oksidaciju aminokiselina ili sintezu bjelančevina 15 - 30%, za razgradnju ugljikohidrata ili njihovo pohranjivanje u glikogen oko 6 - 8% od unijete energije.

Harris-Benedictova jednadžba za određivanje energije bazalnog metabolizma (EBM) uključuje spol, visinu, tjelesnu masu, dob i indeks aktivnosti [7] [8]:

a) za muškarce: $EBM = 66,47 + (13,75 \times \text{tjelesna masa u kg}) + (5,003 \times \text{visina u cm}) - (6,755 \times \text{dob u godinama})$

b) za žene: $EBM = 655,1 + (9,563 \times \text{tjelesna masa u kg}) + (1,850 \times \text{visina u cm}) - (4,676 \times \text{dob u godinama})$,

Harris-Benedictova jednadžba ne uvažava činjenicu da osobe s većom mišićnom masom u strukturi tjelesne mase u odnosu na masno tkivo, kao što su primjerice sportaši, imaju veću EBM. Mišićno tkivo zahtijeva znatno više energije za održavanje zbog povećane razine metaboličke aktivnosti te osobe koje su sportaši imaju veću energiju bazalnog metabolizma (EBM). Suprotno tome, masnom tkivu je potrebno vrlo malo energije za održavanje i malo utječe na odmorne ili bazne potrebe metaboličke energije. Stopa metabolizma u mirovanju i bazalni metabolizam općenito opada s godinama ili ako dolazi do pada tjelesne mase. Aktivnosti koje imaju tendenciju povećanja mišićne mase (mršavog tkiva), kao što su izgradnja tijela ili trening snage (anaerobne aktivnosti), također će povećati bazalni ili odmorni metabolizam. Aerobne aktivnosti kao što su trčanje, klizanje ili skakanje konopa mogu poboljšati izdržljivost, ali imaju mali utjecaj na bazalni ili metabolički stupanj odmora (vidi dohotke nakon vježbanja u nastavku). Ostali čimbenici koji mogu utjecati na EBM ili BBM uključuju stres, bolest, razinu hormona (npr. štitnjača), okoliš (npr. temperaturu ili nadmorsku visinu) ili bilo koji drugi faktor koji utječe na normalno funkcioniranje jednog ili više vitalnih organa. Ukupna dnevna potreba za energijom (UE) (engl. *Total Daily Energy Expenditure* – TDEE) je prosječni unos energije u prehrani za koji se predviđa da će održavati energetska ravnotežu kod zdravih, normalnih pojedinaca određene dobi, spola, težine, visine i razine tjelesne aktivnosti u skladu s dobrim zdravljem i izračunava prema sljedećoj Harris-Benedictovoj jednadžbi:

$$UE \text{ (kcal/dan)} = EBM \times PAL \text{ ili } PA \quad (4 - 2)$$

Gdje je PAL razina tjelesnog napora (engl. *Physical Activity Level* - PAL). U jednadžbi se obično koristi koeficijent tjelesnog napora (engl. *Physical Activity* - PA) koji se kreće u rasponu PAL-a s

obzirom na razinu aktivnosti pojedine dobne skupine žena ili muškaraca (Tablica 4.1) . Razine tjelesnog napora (PAL) definirane su na sljedeći način [9]:

- a) sjedilačke (PAL 1,0 - 1,39) - tipične svakodnevne aktivnosti kao što su vrtlarjenje (bez podizanja), kućni poslovi, lagane aktivnosti dok sjedite, utovarivanje/istovarivanje automobila, košenje, košenje travnjaka (električna kosilica) itd.
- b) slabo aktivne (PAL 1,4 - 1,59) - tipične svakodnevne aktivnosti kao što su vrtlarjenje (bez podizanja), kućanski poslovi, lagane aktivnosti dok sjedite, utovarivanje / istovar automobila, košenje, košenje travnjaka (električna kosilica) itd., uz 30 do 60 minuta umjerenih aktivnosti dnevno kao što su vožnja biciklom (ležerno), golf (bez kolica), plivanje (sporo), hodanje 3-4 km/h itd.
- c) aktivne (PAL 1,6 - 1,89) - tipične svakodnevne aktivnosti kao što su vrtlarjenje (bez podizanja), kućni poslovi, lagane aktivnosti dok sjedite, utovarivanje/istovarivanje automobila, košenje, košenje travnjaka (električna kosilica) itd. uz ≥ 60 minuta umjerene aktivnosti dnevno kao što su vožnja biciklom (ležerno), golf (bez kolica), plivanje (sporo), hodanje 3-4 km/h itd.
- d) vrlo aktivne (PAL 1,9 - 2,5) - tipične svakodnevne aktivnosti kao što su vrtlarjenje (bez podizanja), kućni poslovi, lagane aktivnosti dok sjedite, utovarivanje/istovarivanje automobila, košenje, košenje travnjaka (električna kosilica) itd. uz dodatnih 60 minuta energičnih aktivnosti kao što su sječa drva, penjanje po brdima (bez opterećenja), vožnja biciklom (umjereno), trčanje (10 milja), preskakanje konopa, klizanje, plivanje, tenis (parovi), hodanje (5 km/h) ili 120 minuta dodatne umjerene aktivnosti.

Tablica 4.1. Koeficijenti tjelesnog napora prema dobnim skupinama, spolu i razini tjelesnog napora, izvor: [7] [9]

SPOL I DOB (godina)	SJEDILAČKE	SLABO AKTIVNE	AKTIVNE	VRLO AKTIVNE
dečki 3 – 18	1,00	1,13	1,26	1,42
pretili dečki 3 – 18	1,00	1,12	1,24	1,45
muški > 19	1	1,11	1,25	1,48
djevojčice 3 – 18	1,00	1,16	1,31	1,56
pretile djevojčice 3 - 18	1,00	1,18	1,35	1,60
žene > 19	1,00	1,12	1,27	1,45

5. POSTOJEĆA RJEŠENJA PLANIRANJA ZDRAVE PREHRANE

U ovom dijelu završnog rada usporedit će se već postojeća slična programska rješenja problema zdrave prehrane i realizacija rješenja na Android platformi. Aplikacije će se uspoređivati na temelju određenih prednosti, nedostataka i razlika u rješavanju zadanog problema te na temelju izgleda korisničkog sučelja. Na kraju poglavlja nakon navođenja postojećih aplikacija bit će opisana i osnovna struktura aplikacije *StayHealthy* koja se opisuje u ovom radu.

5.1. Usporedba s postojećim rješenjima

Na mrežnim trgovinama kao što je Google Play dostupnima na Android mobilnim uređajima nalaze se već određene aplikacije koje se bave istim ili sličnim problemom zdrave prehrane. Većina njih omogućuje optimiranje dnevnog unosa kalorija, prehrane, tjelesne mase, a neke i praćenje fizičkih aktivnosti te neke druge naprednije funkcionalnosti. Većina ih je za korištenje dostupna bez naknade, međutim kod nekih su određene funkcionalnosti, primjerice praćenje prehrane za više dana, dostupne samo uz dodatnu novčanu naknadu, što u aplikaciji *StayHealthy* nije slučaj. Prema sličnosti određenih karakteristika izdvojene su tri aplikacije: *Eat This Much*, *MyNetDiary* i *MyHealthPyramid*. Tijekom korištenja navedenih aplikacija utvrđene su sve pozitivne i negativne karakteristike ovih aplikacija koje će biti opisane u nastavku. Iako postoje slične aplikacije dostupne na računalnim platformama te iOS i Windows mobilnim uređajima, navedene su isključivo aplikacije koje se koriste na Android operacijskom sustavu, jer njemu pripada i aplikacija kreirana u ovom radu.

5.1.1. *Eat this much*

Aplikacija *Eat this much* nudi mogućnost besplatnog korištenja uz ograničene funkcionalnosti, ali i nadograđenu *premium* verziju s dodatnim funkcijama. U besplatnoj verziji aplikacija korisniku nudi unos podataka bitnih za izračun dnevnog optimalnog unosa kalorija i planiranje dnevnih obroka, kao što su: tjelesna masa, visina, starost, razina fizičke aktivnosti i ciljna tjelesna masa. Osim ove osnovne funkcionalnosti bitne karakteristike ove aplikacije su [10]: a) automatsko generiranje obroka koji zadovoljavaju korisnikov kalorijski dnevni unos, b) prehrambeni ciljevi koji mogu biti postavljeni za gubitak težine, održavanje težine i izgradnju mišićne mase, c) jela se mogu odabirati iz određenih skupina hrane kao što su vegetarijanska ili mediteranska i d) filtriranje hrane s obzirom na alergije ili želje korisnika. Uz dodatno plaćanje raspoložive su opcije planiranja

obroka za više dana, određivanje prilagođenih ciljeva za svaki dan u tjednu kao što je npr. unos više kalorija na dane kad korisnik trenira te je omogućeno i postavljanje broja članova obitelji u korisničke podatke kako bi se pri nabavi osiguralo dovoljno namirnica za određene obroke.

5.1.2. MyNetDiary

Američki časopis za preventivnu medicinu ocijenio je ovu aplikaciju kao najbolju aplikaciju za prehranu, a također je prepoznat i od američkih dnevnih novina *New York Times* kao jednostavnija, brža i aplikacija boljeg izgleda u usporedbi s drugima [11]. Aplikaciju koristi preko 9 milijuna korisnika diljem svijeta koji ostvaruju dobre rezultate u smanjivanju tjelesne mase. Aplikacija osim osnovnih funkcija planiranja prehrane s obzirom na određeni kalorijski unos sadrži brojne savjete, preporuke i recepte. Korisniku omogućuje unošenje svojih recepata i sastavljanje liste omiljene hrane. Također, aplikacija sadrži veliku raznolikost u bazi podataka s hranom iz raznih restorana, trgovina, specijalnih dijeta i karakterističnih za određeno područje. Za razliku od aplikacije *Eat this much* ova aplikacija omogućuje sastavljanje plana prehrane za bilo koji datum koji korisnik odabere. Aplikacija nije jednostavna za korištenje posebice novim korisnicima, jer je složena i opterećena dodatnim funkcijama kao što su spremanje slika prije i poslije planiranja prehrane, dnevne zabilješke, razni grafovi i slično.

5.1.3. MyHealthPyramid

MyHealthPyramid je jednostavna korisnička aplikacija koja se koristi za postavljanje i bilježenje svakodnevnih prehrambenih navika, vježbanja i održavanja hidratacije [12]. Praćenje prehrane u ovoj aplikaciji oslanja se na omjere pojedine vrste hrane prema američkoj prehrambenoj piramidi, ali nema opciju s personaliziranim optimiranjem kalorijskog unosa. Aplikacija sadrži dnevnik koji pokazuje broj i vrstu obroka ovisno o grupi hrane i vrijeme utrošeno na vježbanje te izdaje upozorenja u slučaju prekomjernog broja obroka određene kategorije hrane, primjerice slatkiša koji su na vrhu prehrambene piramide. Ova aplikacija ne sadrži baze podataka s hranom i korisna je samo za ljude koji žele imati uvid u vrste obroka i način života u određenom razdoblju, ali neće koristiti ljudima koji imaju značajnijih problema s tjelesnom masom zbog koje im je potrebna detaljnija analiza i organizacija prehrane.

5.1.4. Usporedba s prethodno nabrojanim aplikacijama

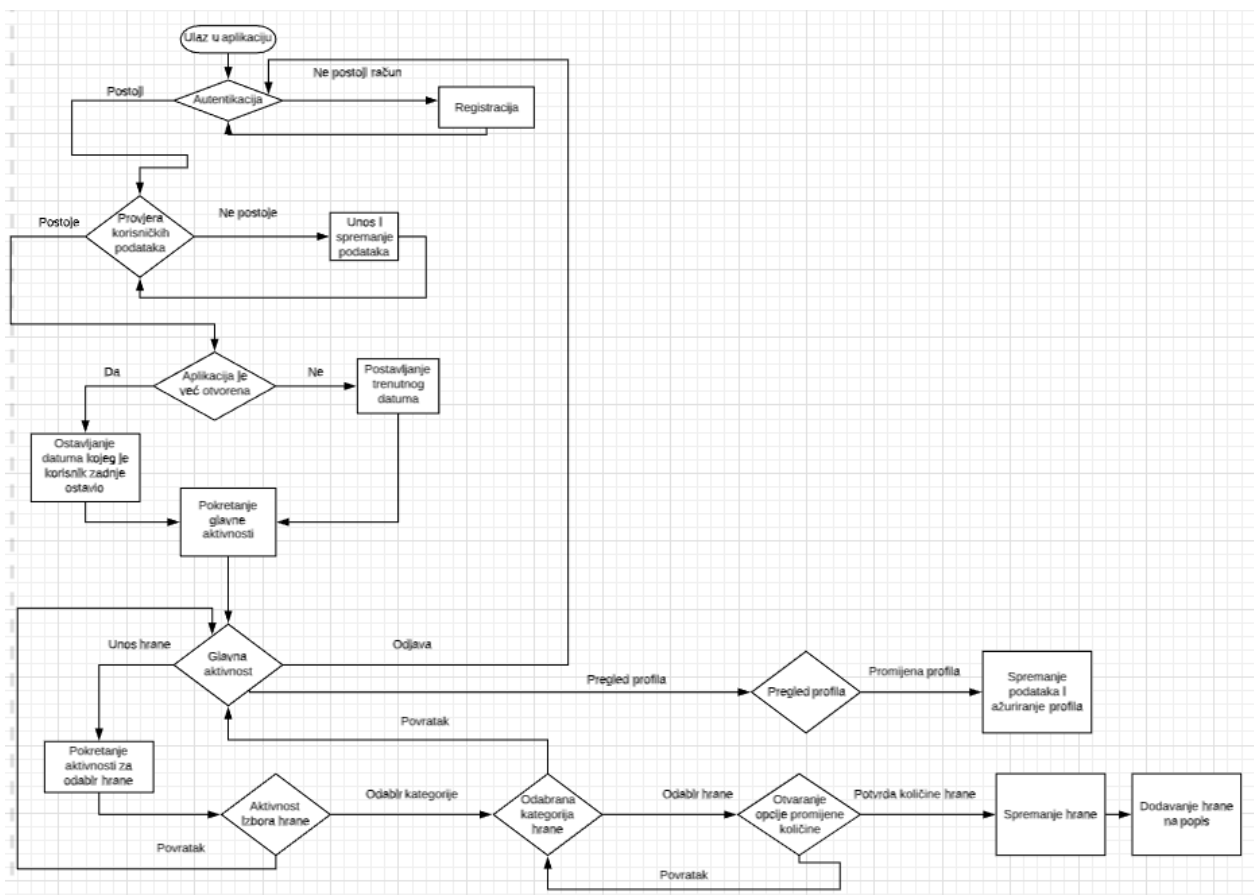
Aplikacija kreirana u ovom radu ima neka obilježja prethodno nabrojanih aplikacija, ali i neke različitosti i možebitne nedostatke. Zdrava prehrana kojom se vodi ova aplikacija slična je kao u aplikaciji *MyHealthPyramid*, ali sa dodatnim funkcionalnostima i detaljima. Konceptcija prehrane i baza podataka s hranom usklađena je s najnovijom američkom verzijom prehrambene piramide iz 2011. godine *MyPlate*. Aplikacija ima opciju odabira datuma za razliku od *Eat This Much* aplikacije koja ovu opciju ima tek uz dodatno plaćanje. Glavna prednost aplikacije *StayHealthy* u odnosu na *MyNetDiary* aplikaciju je jednostavnost korištenja koja novim korisnicima omogućuje lakše i brže pretraživanje i biranje opcija. Aplikaciju je potrebno dodatno razvijati prije svega u smjeru povećanja baze podataka hrane koju korisnik može odabirati.

5.2. Koncept aplikacije *StayHealthy*

Nakon stvaranja teorijske podloge vezane za zdravu prehranu potrebno je strukturirati koncept programskog rješenja koji će analizirati određene potrebe i zahtjeve korisnika i na najbrži način omogućiti njihovo ispunjavanje. Uslijed ulaska u aplikaciju korisnik odabire opciju registracije ako ranije nije kreirao račun ili opciju prijave ako se već prethodno registrirao. Autentifikacija se vrši preko e-mail adrese i lozinke iste te adrese i na taj način je omogućena korisniku lakša prijava i manipulacija identifikacijskim podacima u slučaju promjene ili zaboravljanja lozinke. Nakon prijavljivanja u sustav u bazi podataka provjerava se postojanje korisničkih podataka potrebnih za izračune opisane u prethodnom poglavlju. Ako su podaci već prethodno kreirani i pohranjeni, korisniku se odmah otvara glavna aktivnost na kojoj se nalazi glavne funkcionalnosti aplikacije. U suprotnom, korisnik mora ispuniti obrazac sa svojim podacima i potvrditi ih kako bi se podaci spremili i kako bi se izvršili izračuni. Osim stvaranja profila korisnika, pri ulasku u aplikaciju program vrši provjeru datuma kako bi se postavio trenutni datum na glavnom dijelu aplikacije koji korisnik ne mora nužno koristiti, nego može odabirati po izboru datume za koje želi napraviti plan prehrane. Sve dok se aplikacija ne zatvori, pri svakom ulasku u aplikaciju datum će biti postavljen na zadnji datum koji je korisnik postavio. Na glavnoj aktivnosti programa može se sastavljati plan prehrane za dane po izboru korisnika te se na toj aktivnosti korisniku pokazuje koliki bi trebao biti ukupan unos kalorija za taj dan, ali i posebno unos kalorija za pojedinu kategoriju hrane. Korisnik može po izboru birati i pretraživati hranu po kategorijama i dodavati željenu količinu. U slučaju prekoračenja zdravog omjera hrane korisnika će se na adekvatan način upozoriti. Osim dodavanja, mijenjanja i brisanja hrane za pojedine dane, korisnik ima mogućnost promijeniti i pohraniti nove

podatke svog profila i tako promijeniti i ciljeve prehrane. Na istom dijelu aplikacije gdje mu je omogućeno mijenjanje profila, korisnik ima uvid u izračunati indeks tjelesne mase i prema njemu izračunati unos kalorija potreban za njegovu optimizaciju.

Sve podatke koje korisnik unosi u aplikaciju, a vezane su za izbor hrane za određene datume i promjenu korisničkog profila, automatski se pohranjuju u bazu podataka te se ti podaci generiraju prilikom svakog ulaska korisnika u aplikaciju. U aplikaciji postoji još i dio u kojem se nalaze savjeti koji korisniku predstavljaju pomoć za što bolje razumijevanje pojma zdrave prehrane. Na slici 5.1 prikazan je blokovski dijagram koji opisuje koncept ove aplikacije.



Slika 5.1. Dijagram funkcija aplikacije

6. KORIŠTENA TEHNOLOGIJA

U ovom poglavlju bit će opisane tehnologije i koncepti koji se koriste za izradu ove aplikacije. Navest će se glavna obilježja operacijskog sustava Android, razvojne okoline te programi i programski jezici korišteni za pisanje koda. Upoznavanje s teorijom osnovnih programskih komponenata obavlja se kako bi se lakše mogao pratiti proces razvoja aplikacije. Aplikacija, osim što sadrži programski kod koji obavlja interakciju s korisnikom i prikazuje mu jednostavno i razumljivo sučelje, ona izvršava veliku količinu koda koji sprema podatke u odgovarajuću bazu podataka i manje dinamičke spremnike, a sve se odvija u pozadini aplikacije bez opterećivanja korisnika.

6.1. Razvojna okolina Androida

Ova aplikacija namijenjena je isključivo za Android mobilne uređaje, kao što su to pametni telefoni i tablet računala, stoga je vrlo bitno znati koje sve mogućnosti i načine programiranja ima ova okolina te nakon tih spoznaja potrebno je odabrati najbrži i najefikasniji način.

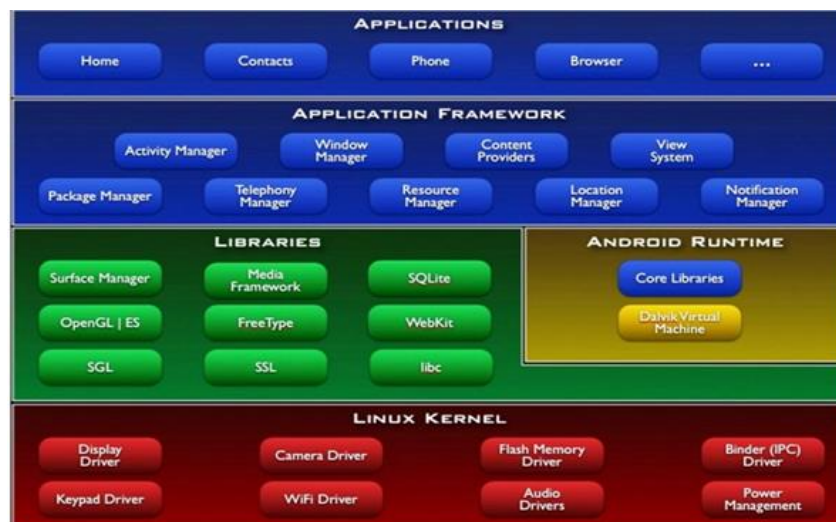
6.1.1. Općenito o operacijskom sustavu Androida

Android je operacijski sustav temeljen na jezgri Linux-a, razvijen uglavnom za mobilne uređaje sa zaslonom osjetljivim na dodir, kao što su to pametni telefoni i tablet računala. Nakon nekog vremena operacijski sustav prilagođen je i za pametne televizore, za automobilska sredstva te za pametne satove. Svaki od tih softvera prilagođen je jedinstvenom korisničkom sučelju. Android je programska podrška koja je osnovana 2003. godine u Kaliforniji gdje ju je u početku razvijala tvrtka Android Inc., ali 2005. godine tu tvrtku preuzima Google. Ovaj operacijski sustav prvi put je predstavljen javnosti 2007. godine, a nedugo zatim u rujnu 2008. godine pojavljuju se i prvi uređaji bazirani na Androidu. Kako je od samih početaka Android zamišljen kao sustav otvorenog koda cijeli kod dostupan je pod *Apache* licencom od listopada iste godine kad su se pojavili prvi uređaji. Google razvija paket vlasničkog softvera koji je povezan s Androidom, a uključuje aplikacije za razne servise, primjerice Gmail ili Google karte, te trgovinu aplikacija Google Play. Od 2011. godine Android je najprodavaniji mobilni operacijski sustav za pametne telefone, a za tablete od 2013. godine. Dokaz tomu je u samoj činjenici da ovaj operacijski sustav ima više od 2 milijarde mjesečno aktivnih korisnika, a Google Play trgovina obuhvaća više od 3,5 milijuna aplikacija. Android zahvaljujući svom otvorenom izvornom kodu omogućava mnogobrojne

beneficije kao što su povezivanje s drugim uređajima pomoću raznih komunikacijskih tehnologija (primjerice Bluetooth, LTE, UMTS i slično), potporu za povezivanje uređaja za upravljanje (tipkovnica, kompjuterski miš itd.), slanje i primanje poruka za koje se koriste SMS i MMS servisi, internetski preglednik, bežično povezivanje na internetsku mrežu i mnoge druge. Unatoč tome što su ta dva programska jezika primijenjeni za razvoj radnog okružja, odnosno *framework*, i što se uz njih mogu koristiti i drugi programski jezici za kodiranje u Androidu, primjerice Kotlin, veći dio aplikacija, kao i aplikacija samog ovog rada pisana je u Java programskom jeziku koristeći *Android Software Development Kit (SDK)*.

6.1.2. Arhitektura Androida

Arhitektura Androida sastoji se od nekoliko programskih razina kao što se vidi na slici 6.1.



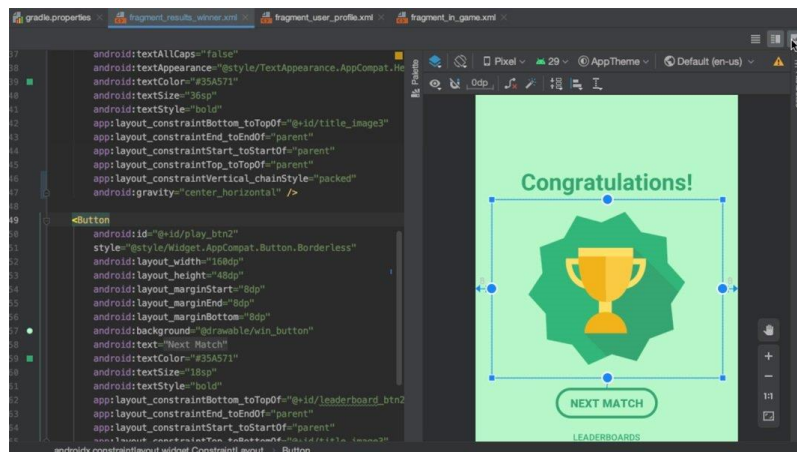
Slika 6.1: Prikaz dijelova Android OS-a, izvor: [13]

Pri dnu se nalazi Linux jezgra koja sadrži brojne upravljačke programe (drivere), od kojih se ističu driver za međuprocesnu komunikaciju koji se koristi za razmjenu podataka između različitih procesa te upravljački program za napajanje. Odmah iznad jezgre nalaze se biblioteke koje sadržavaju određene funkcionalnosti, a neke od njih su SQLite, WebKit, SGL i druge. Nakon toga sljedeći sloj je Android *Runtime* koji se koristi za pokretanje aplikacija, a sastoji se od dva dijela: biblioteke koje sadrže većinu jezgrenih biblioteka programskog jezika Java te *Dalvik Virtual Machine* koji transformira Java *class* datoteke u vlastiti format *.dex* kako bi se osigurao minimalni

trošak memorije. Nakon toga slijedi aplikacijski okvir (engl. *Application Framework*) koji omogućava kreiranje aplikacija, odnosno upravlja raznim njenim komponentama kao što su resursi, programski paketi, dohvaćanje lokacija korisnika i slično. Na vrhu se nalazi sloj aplikacija. To je sloj vidljiv krajnjem korisniku i čine ga ugrađene aplikacije kao što su SMS programi, kalendari, web preglednici te aplikacije korisnika koje se mogu naći na online trgovini.

6.2. Android Studio

Android Studio službeno je integrirano razvojno okruženje za Googleov operacijski sustav Android, izgrađen na JetBrains-ovom IntelliJ IDEA *software*-u i dizajniran posebno za razvoj Androida [14]. O tome je javnost prvi put informirana 16. svibnja 2013. godine na Google I/O konferenciji [15]. Međutim, prva stabilna struktura objavljena je u prosincu 2014. godine s početnom verzijom 1.0. Dostupan je za instalaciju na softverima koji se temelje na Windows, macOS i Linux operacijskim sustavima, ali od 2020. dostupan je i kao usluga s pretplatom. Primarni programski jezik koji se koristio za razvoj aplikacija u ovom okruženju bio je obično Java koja je i danas podržana i često korištena, ali od 2019. Google za razvoj Android aplikacija preferira Kotlin. U trenutnoj verziji Android Studia neke od dostupnih funkcionalnosti su Gradle podrška za izgradnju, alati za refaktoriranje i brze popravke, alati za održavanje performansi, upotrebljivosti i kompatibilnosti verzija, stvaranje raznih korisničkih sučelja uz pomoću uređivača rasporeda (engl. *layout editor*) koji dopušta i *drag and drop* postavljanje elemenata, mogućnost povezivanja s Firebase *Cloud* platformom za razmjenu poruka i obavijesti među Android uređajima, pokretanje aplikacije na virtualnom uređaju te druge funkcionalnosti. Na slici 6.2 vidljiv je izgled razvojnog okruženja Android Studio-a.



Slika 6.2. Razvojno okruženje Android Studio

6.2.1. Android SDK

Android SDK (engl. *Software Development Kit*) skup je razvojnih alata koji služe za razvoj aplikacija na Android platformi [16]. On uključuje razne potrebne biblioteke, program za ispravljanje grešaka, emulator, raznu dokumentaciju za izgradnju sučelja, primjerke raznih kodova te tutorijale koji podučavaju o Android operacijskom sustavu. Odgovarajući SDK izdaje se svaki puta kad Google izdaje novu verziju Androida. Iako je moguće pisanje Android programa u naredbenom retku (engl. *Command prompt*) najčešće se koristi integrirano razvojno okruženje (IDE) koje nudi grafičko sučelje i na taj način omogućuju programerima brže izvršavanje određenih razvojnih ciljeva. Ako se pišu aplikacije uz pomoć Java programskog jezika potrebno je imati instaliran i *Java Development Kit* (JDK) koji sadrži različite programske alate potrebne za razvoj aplikacije.

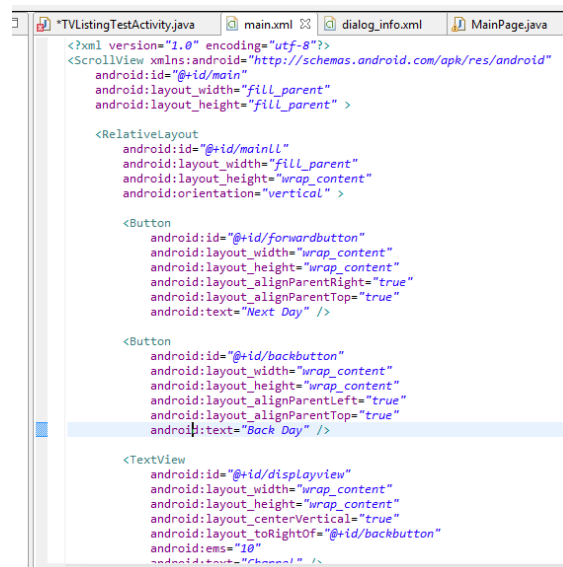
6.3. Java programski jezik

Java je objektno orijentirani programski jezik koji je kreirala tvrtka Sun Microsystems od kojih se posebno ističu inženjeri James Gosling i Patrick Naughton. Objektno orijentirani programski jezik je jedan od pristupa programiranju u kojem se projektirana aplikacija strukturira pomoću klasa i njihovih konkretnih realizacija, objekata, koji međusobno komuniciraju i izvršavaju određene funkcije. Prva verzija ovog programskog jezika objavljena je u studenom 1995. i ono po čemu se razlikovao od dotadašnjih programskih jezika je to što se aplikacije pisane u Javi mogu izvoditi bez modifikacija na svim operacijskim sustavima koji imaju instaliran JVM (engl. *Java Virtual Machine*) razvijen za određeni operacijski sustav dok je za neke starije programe, primjerice C programski jezik, potrebno prilagođavati kod operacijskom sustavu na kojem se izvodi. Do 2018. Java programski jezik je odmah iza JavaScript bio najzastupljeniji jezik na najpoznatijoj platformi koja pruža usluge pohranjivanja kodova i ostalih podataka, a naziva se GitHub [17]. Java je i danas jedan od najkorištenijih programskih jezika zbog svoje jednostavnosti, lakoće učenja, besplatnog korištenja, brzine i sigurnosti. Neke od statistika koje se bave brojem korisnika određenih programskih jezika kazuju da se broj korisnika ovog jezika kreće između 7 do više od 10 milijuna, a broj uređaja koji pokreću Java kod se kreće oko 3 milijarde [18]. Na sadašnjem programerskom tržištu, Java se koristi u širokom području programskog inženjerstva pogotovo tamo gdje prevladava važnost brzine razvijanja programskog sustava nad zahtjevima za brzinom rada programa. Općenito se smatra da se na Javi može brže razviti program zbog velike zastupljenosti

na tržištu i velike količine gotovih kodova te da se razvoj obavlja s puno manje grešaka nego u nekim srodnim jezicima. Prvenstveno zbog toga je popularna za razvoj aplikacija na mobilnim uređajima, kod programa finansijskih poduzeća, razvoj aplikacija za radnu površinu, razvoj različitih igara, povezivanje baza podataka i još veliki broj drugih programskih aktivnosti.

6.4. XML

XML (engl. *Extensible Markup Language*), predstavlja standardiziran jezik za označavanje podataka za čiju se standardizaciju brine konzorcij *World Wide Web*-a. Vizija kojom su se programeri vodili dok su razvijali ovaj jezik je bila da stvore jezik koji će biti jednostavno razumljiv i čitljiv ljudima, ali i računalnim programima u bilo kojem programu za uređivanje te jezik koji će biti dovoljno fleksibilan da se, uzimajući u obzir zahtjeve, može proširivati novim oznakama za opisivanje stranica ili sadržaja datoteka. Prvu verziju XML jezika objavio je konzorcij *World Wide Web*-a 1998. te je do današnjeg dana taj jezik korišten u širokom području programiranja i koristi se za razne potrebe: razdvajanje podataka od prezentacije, razmjenu podataka na webu, spremanje podataka, izradu novih jezika za označavanje i druge slične funkcije. Temeljna građevna jedinica XML dokumenta je element opisan oznakama. Naziv elementa obično opisuje njegov sadržaj, a struktura opisuje odnos između elemenata. XML ima dodatnu mogućnost definiranja atributa za elemente u početnoj oznaci elementa kako bi se detaljnije opisale karakteristike istih. Na slici 6.3 vidi se programski kod pisan u XML jeziku.



```
<?xml version="1.0" encoding="utf-8"?>
<ScrollView xmlns:android="http://schemas.android.com/apk/res/android"
    android:id="@+id/main"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent" >

    <RelativeLayout
        android:id="@+id/mainLL"
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:orientation="vertical" >

        <Button
            android:id="@+id/forwardbutton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentRight="true"
            android:layout_alignParentTop="true"
            android:text="Next Day" />

        <Button
            android:id="@+id/backbutton"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_alignParentLeft="true"
            android:layout_alignParentTop="true"
            android:text="Back Day" />

        <TextView
            android:id="@+id/displayview"
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:layout_centerVertical="true"
            android:layout_toRightOf="@+id/backbutton"
            android:ems="10"
            android:text="Channel" />

    </RelativeLayout>

</ScrollView>
```

Slika 6.3. XML datoteka

6.5. Firebase

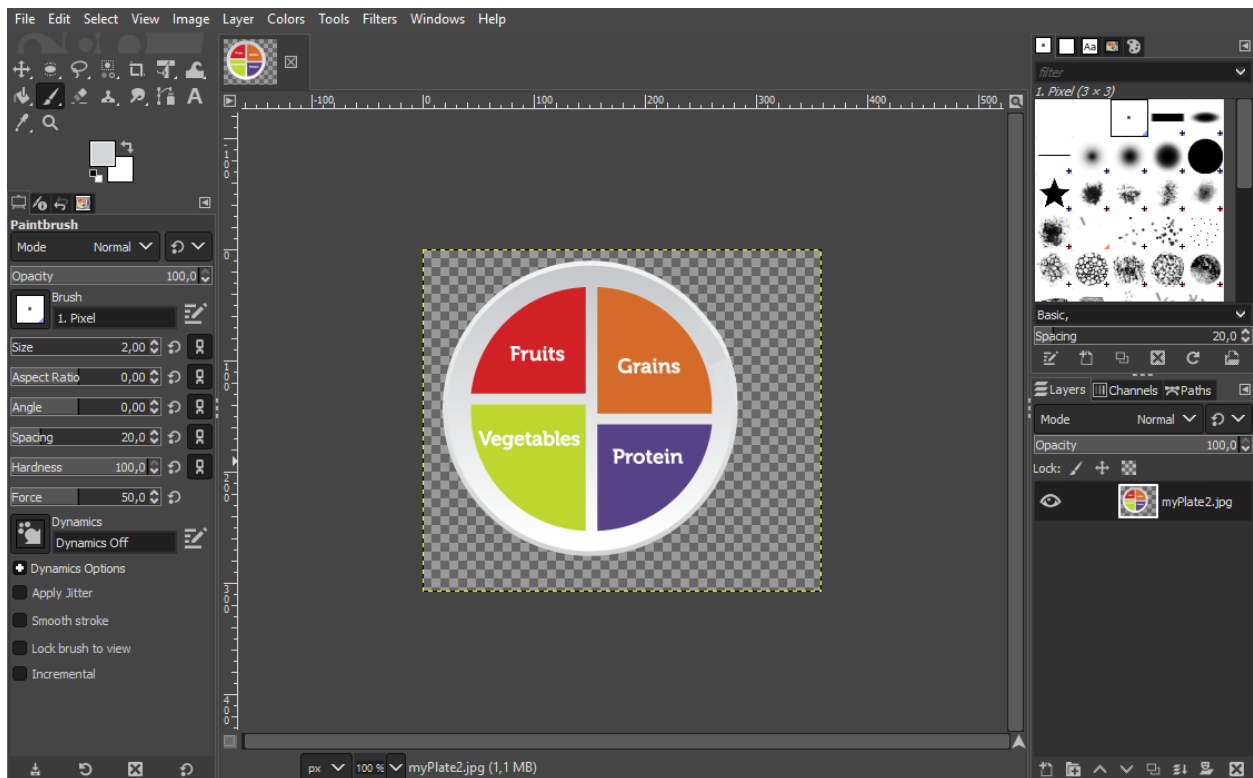
Firebase je platforma za razvoj mobilnih i web aplikacija koju je razvio Firebase Inc. 2011. godine, a od 2014. je u vlasništvu Google [19]. Od 2020. Firebase koristi više od 1,5 milijuna aplikacija. Prvi proizvod Firebase-a bila je baza podataka za rad u stvarnom vremenu (engl. *Firebase Realtime database*) koja obuhvaća aplikacijsko programsko sučelje koje sinkronizira podatke na raznim platformama kao što su iOS, Android i web uređaji te ih pohranjuje uz mogućnost dijeljenja podataka među različitim korisnicima. 2017. godine tvrtka je pokrenula *Cloud Firestore* koji predstavlja bazu podataka u stvarnom vremenu temeljenu na dokumentima i kolekcijama te je nasljednik izvorne baze podataka za rad u stvarnom vremenu.

Firebase je skup alata koji pokriva veliki dio usluga i zadataka koje programeri obično moraju sami kreirati, a uz pomoć ove platforme mogu se usredotočiti na druge funkcionalnosti svoje aplikacije. Te usluge uključuju razne analitike, autentifikaciju, baze podataka, konfiguracije, pohrane datoteka, slanje notifikacija te razne druge pomoćne usluge. Podaci, odnosno usluge koje se pohranjuju, smještaju se u oblak s vrlo malo napora programera gdje se Firebase ponaša kao domaćin te na siguran način održava i upravlja pohranjenim podacima i funkcionalnostima aplikacije. S Firebase proizvodima programeri mogu preskočiti razvijanje *backenda* i ostaviti da taj posao napravi ova platforma. S Android aplikacijama Firebase se povezuje preko biblioteka i koristi se preko različitih referenca ovisno o uslugama kojima se koristi. Za aplikaciju *StayHealthy* će se osim standardne *Firebase Realtime Database* koristiti i zadnja verzija *Cloud Firestore* koji omogućuje istu funkcionalnost, ali za razliku od prethodne verzije čiji se prikaz predstavlja u obliku hijerarhije stabla, omogućuje prikaz hijerarhije u obliku kolekcija, dokumenata i polja što olakšava njezino korištenje. Također, koristit će se i *Firebase Authentication* koji će omogućavati prijavu preko e-mail adrese radi olakšanja registracije i prijave korisnika te rješavanja problema promjene lozinke.

6.6. GIMP

GIMP (engl. *GNU Image Manipulation Program*) je najpopularniji besplatni program otvorenog koda za kreiranje i obradu digitalne slike. Mogućnosti obuhvaćene u ovom programu su retuširanje, uređivanje i mijenjanje sastava slike te izvršavanje još mnogih drugih specijaliziranih grafičkih zadataka. Aplikacija je dostupna na raznim platformama kao što su Linux, Windows, OS X i drugi operacijski sustavi. Spencer Kimball i Peter Mattis su izvorno objavili GIMP, a razvoj

je počeo 1995. kao studentski projekt u Kaliforniji, dok je prva verzija objavljena je u siječnju 1996. [20]. Kako je to besplatni program otvorenog koda moguće je mijenjati izvorni kod te poslije te iste promijene i distribuirati. Neke funkcionalnosti ovog programa su: korištenje kompletnog skupa alata za slikanje, uzimanje uzoraka piksela boje koja se onda može koristiti u daljnjoj obradi slike, neograničen broj otvorenih slika odjednom, transformacijski alati, alati za odabir u raznim oblicima, korištenje animacija te rukovanje različitim vrstama datoteka kao što su bmp, pdf, png i mnoge druge. GIMP dopušta čak i vanjske skripte koje može pokretati i na taj način omogućiti dodatne funkcije za obradu slika. Ovaj program koristio se unutar aplikacije *StayHealthy* za određene modifikacije i izradu slika, a ponajprije za sliku na glavnoj aktivnosti aplikacije prikazanoj na slici 6.4.



Slika 6.4. Uređivanje glavne slike aplikacije *StayHealthy*

7. RAZVOJ APLIKACIJE *STAYHEALTHY*

Aplikacija *StayHealthy* ima složenu strukturu od puno kompleksnih dijelova, stoga će se u nastavku teksta opisati najvažniji dijelovi koda koji su bitni za razumijevanje, korištenje i proces razvoja ove aplikacije. Programski kod raspoređen je na više klasa, aktivnosti, fragmenata i uređivača izgleda te se koristi veliki dio funkcionalnosti dostupnih u Android razvojnom okruženju kao što su *RecyclerView*, dijaloški okviri, *on line* baze podataka za pohranu kategorija hrane i planova prehrane, kalendari i slično.

7.1. Gradle

Strukturu gradnje Android Studio temelji na Gradle-u koji omogućuje konfiguraciju izgradnje aplikacije i može se pokretati i ažurirati neovisno o Android Studio-u [21]. Sustav gradnje Gradle u Android studio-u olakšava uvrštavanje vanjskih binarnih zapisa ili drugih modula biblioteke u svoju izgradnju kao zavisnosti. Svaka od tih zavisnosti (engl. *dependencies*) zahtjeva različitu vrstu biblioteke i njihovih funkcionalnosti. Na slici 7.1 vidljive su implementacije ovisnosti biblioteka potrebne za izgradnju ovog projekta. Neke od njih su ovisnosti za uključivanje Firebase i Firestore baze podataka i nekih njihovih funkcija, uključivanje *RecyclerView-a*, ovisnosti potrebne za korištenje navigacijske aktivnosti i ostalo.

```
dependencies {
    implementation fileTree(dir: 'libs', include: ['*.jar'])
    implementation 'androidx.appcompat:appcompat:1.2.0'
    implementation 'androidx.constraintlayout:constraintlayout:1.1.3'
    implementation 'androidx.legacy:legacy-support-v4:1.0.0'
    implementation 'com.google.android.material:material:1.0.0'
    implementation 'androidx.navigation:navigation-fragment:2.0.0'
    implementation 'androidx.navigation:navigation-ui:2.0.0'
    implementation 'androidx.recyclerview:recyclerview:1.1.0'
    testImplementation 'junit:junit4:1.2'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.1.0'
    androidTestImplementation 'androidx.test.espresso:espresso-intent:3.1.0'

    //for firebase and editing button etc.
    implementation 'com.google.firebase:firebase-analytics:17.0.2'
    implementation 'com.cerheuen.elegant-number-button:lib:1.0.2'

    implementation 'com.google.firebase:firebase-core:17.0.2'
    implementation 'com.google.firebase:firebase-database:19.0.0'
    implementation 'info.hoangsf:fbutton:1.0.5'
    implementation 'com.rengwuxian.materialedittext:library:2.1.4'
    // added from login and registration
    implementation 'com.google.firebase:firebase-auth:19.0.0'

    def recyclerview_version = "1.1.0"
    implementation "androidx.recyclerview:recyclerview:$recyclerview_version"

    implementation 'com.squareup.picasso:picasso:2.5.2'
    implementation 'com.squareup.retrofit2:retrofit:2.5.0'
    implementation 'com.squareup.retrofit2:converter-gson:2.4.0'
    implementation 'com.firebaseui:firebase-ui-database:1.2.0'
    implementation 'androidx.coordinatorlayout:coordinatorlayout:1.1.0'
    implementation 'com.github.bumptech.glide:glide:4.9.0'
    annotationProcessor 'com.github.bumptech.glide:compiler:4.9.0'
    implementation 'com.google.firebase:firebase-core:17.0.2'
    implementation 'com.google.firebase:firebase-database:19.0.0'
    implementation 'info.hoangsf:fbutton:1.0.5'
    implementation 'com.rengwuxian.materialedittext:library:2.1.4'
    implementation 'com.android.support:cardview-v7:27.1.1'
    implementation 'com.google.android.support:design:27.1.1'
    implementation 'com.squareup.picasso:picasso:2.5.2'
    implementation 'com.firebaseui:firebase-ui-database:1.2.0'
    implementation 'com.google.firebase:firebase-firestore:19.0.0'
    implementation 'com.firebaseui:firebase-ui-firestore:4.1.0'
}
```

Slika 7.1. Gradle datoteka

7.2. AndroidManifest datoteka

Svaki Android projekt ima AndroidManifest.xml datoteku koja predstavlja korijen projekta. Ta datoteka sadrži važne informacije o aplikaciji: od Android alata za izgradnju, ime paketa aplikacije, informacije o operacijskom sustavu i Google Play trgovini. Osim tih podataka u njoj su navedene komponente, odnosno aktivnosti i servisi koji čine projekt te razne dozvole koje su aplikaciji nužne za pristup zaštićenim segmentima sustava, sadržaju aplikacije i drugim aplikacijama. U ovoj aplikaciji pristupa se internetskim bazama podataka pa je zbog toga implementirano dopuštenje za pristup internetu. Osim te modifikacije unutar ove datoteke mijenjao se i logotip aplikacije vidljiv na slici 7.2. koji čini ovu aplikaciju prepoznatljivom od strane korisnika.



Slika 7.2. Logotip aplikacije

7.3. Aktivnost *MainActivity*

Prema definiciji službene stranice Android programera aktivnost je: „jedinstvena i usredotočena stvar koju korisnik može učiniti“ [22]. Aktivnosti se često poistovjećuju s jednim zaslonom na ekranu koji je u interakciji s korisnikom. Svaka aktivnost se brine za kreiranje prozora te se postavlja sučelje s kojim korisnik vrši interakciju. Međutim, osim aktivnosti koja se pojavljuje na zaslonu interakcija se može vršiti i preko drugih elemenata kao što su to fragmenti ili dijaloški okviri. Početna aktivnost koja se pojavljuje sadrži dva gumba od kojih jedan služi za otvaranje nove aktivnosti na kojem je moguće napraviti registraciju i drugi gumb koji služi za prijavu u

aplikaciju ako se korisnik već prethodno registrirao. Na slici 7.3. vidljiv je izgled aktivnosti *MainActivity*.



Slika 7.3. Aktivnost *MainActivity*

7.4. Aktivnost *SignUp*

Ako korisnik nije ranije registriran u sustavu baze podataka potrebno je obaviti registraciju na aktivnosti *SignUp*. Za registraciju i općenitu kontrolu prijavljenih korisnika koristi se referenca na `FirebaseAuth` koja je još jedna od usluga Firebase platforme. Na slici 7.4. prikazana je metoda kojom se kreira novi korisnik u kojoj se nalaze i određene provjere i uvjeti koji moraju biti zadovoljeni, a to su da okviri za email i lozinku ne budu prazni te da lozinka odgovara email adresi korisnika. Aktivnost za prijavu *LogIn* je slična aktivnosti *SignUp* kako izgledom tako i programskim kodom pa ju nije potrebno dodatno opisivati.

```
btnSignUp.setOnClickListener((v) - {
    String email = etEmail.getText().toString();
    String password = etPasworrdd.getText().toString();

    if(email.isEmpty()) {
        etEmail.setError("Please enter your email");
        etEmail.requestFocus();
    }

    else if(password.isEmpty()){
        etPasworrdd.setError("Please enter your password");
        etPasworrdd.requestFocus();
    }

    else if (email.isEmpty() && password.isEmpty()) {
        Toast.makeText( context SignUp.this, text "Fields are empty!", Toast.LENGTH_SHORT).show();
    }

    else if(!email.isEmpty() && password.isEmpty()) {
        mFirebaseAuth.createUserWithEmailAndPassword(email,password).addOnCompleteListener( activity: SignUp.this, new OnCompleteListener<AuthResult>() {
            @Override
            public void onComplete(@NonNull Task<AuthResult> task) {
                if(!task.isSuccessful()) {
                    Toast.makeText( context: SignUp.this, text: "SignUp Unsuccessful, please try again", Toast.LENGTH_SHORT).show();
                }

                else {
                    startActivity(new Intent( packageContext: SignUp.this, ParameterActivity.class));
                }
            }
        });
    }

    else {
        Toast.makeText( context: SignUp.this, text: "Error occurred!", Toast.LENGTH_SHORT).show();
    }
});
});
```

Slika 7.4. Registracija korisnika u aktivnosti *SignUp*

7.5. Aktivnost *ParameterActivity*

Ova aktivnost otvara se samo u slučaju ako korisnik dotada nije ispunjavao obrazac koji je na aktivnosti prikazan. Obrazac prema slici 7.5., osim osnovnih podataka poput imena i prezimena korisnika, potražuje sve podatke koji su potrebni za računanje indeksa tjelesne mase i potrebnog kalorijskog unosa korisnika.

Slika 7.5. Aktivnost *ParameterActivity*

Podaci vezani za korisnika unutar aplikacije se spremaju u klasu *User* koja predstavlja model pomoću kojeg se lakše upravlja podacima korisnika te omogućuje lakšu pohranu u bazu podataka. Prema slici 7.6. taj model sadrži konstruktor i metode kojima se postavljaju i dohvaćaju podaci.

```
public class User {  
  
    private String Name;  
    private String Gender;  
    private int Age;  
    private int Height;  
    private int Weight;  
    private String ActivityLevel;  
  
    public User() {  
    }  
  
    public User(String Name, String Gender, int Age, int Height, int Weight, String ActivityLevel) {  
        this.Name = Name;  
        this.Gender = Gender;  
        this.Age = Age;  
        this.Height = Height;  
        this.Weight = Weight;  
        this.ActivityLevel = ActivityLevel;  
    }  
  
    public String getName() { return Name; }  
    public void setName(String name) { Name = name; }  
    public String getGender() { return Gender; }  
    public void setGender(String gender) { Gender = gender; }  
    public int getAge() { return Age; }  
    public void setAge(int age) { Age = age; }  
    public int getHeight() { return Height; }  
    public void setHeight(int height) { Height = height; }  
    public int getWeight() { return Weight; }  
    public void setWeight(int weight) { Weight = weight; }  
    public String getActivityLevel() { return ActivityLevel; }  
    public void setActivityLevel(String activityLevel) { ActivityLevel = activityLevel; }  
}
```

Slika 7.6. Prikaz klase *User*

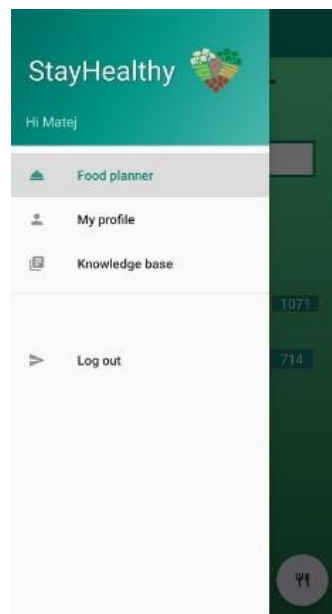
Pohrana podataka u Firestore bazu podataka obavlja se pritiskom na gumb „*CONFIRM DATA*“ preko reference koja dohvaća kolekciju *users*, a potom i dokument s jedinstvenim *ID*-em korisnika generiranom od strane baze. Referenca pozivom metode *addOnSuccessListener* pohranjuje podatke korisnika u bazu i započinje novu aktivnost što je vidljivo na slici 7.7.

```
userID = FirebaseAuth.getCurrentUser().getUid();  
DocumentReference documentReference = Firestore.collection(collectionPath "users").document(userID);  
  
User mUser = new User(mName.getText().toString(), Gender.toString(), Integer.valueOf(mAge.getText().toString()),  
    Integer.valueOf(mHeight.getText().toString()), Integer.valueOf(mWeight.getText().toString()),  
    String.valueOf(spinnerActivityLevel.getSelectedItem()));  
Map<String, Object> user = new HashMap<>();  
user.put("Name", mUser.getName());  
user.put("Gender", mUser.getGender());  
user.put("Age", mUser.getAge());  
user.put("Height", mUser.getHeight());  
user.put("Weight", mUser.getWeight());  
user.put("Activity level", mUser.getActivityLevel());  
  
documentReference.set(user).addOnSuccessListener(new OnSuccessListener<Void>() {  
    @Override  
    public void onSuccess(Void aVoid) {  
        Toast.makeText(context, ParameterActivity.this, text: "User profile is successfully created", Toast.LENGTH_SHORT).show();  
    }  
}).addOnFailureListener((e) -> {  
    Log.d(tag: "ParameterActivity", msg: "User parameters failure" + e.toString());  
});  
  
startActivity(new Intent(context, ParameterActivity.this, HomeActivity.class));
```

Slika 7.7. Pohrana podataka korisnika u bazu podataka

7.6. Aktivnost *HomeActivity*

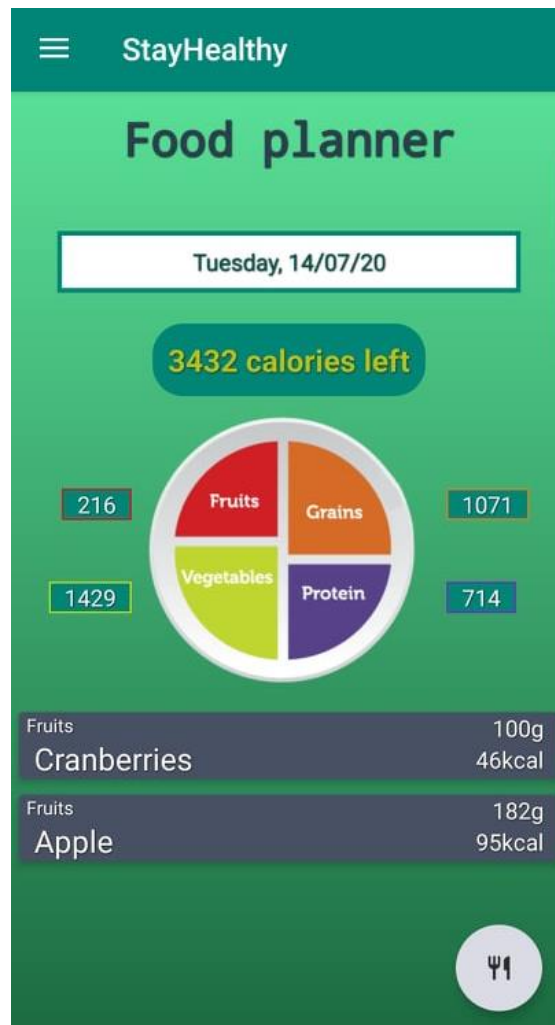
Aktivnost *HomeActivity* najvažnija je aktivnost u aplikaciji. Prema slici 7.8. aktivnost koristi navigacijsku ladicu (engl. *Navigation drawer*) pomoću koje može navigirati na određene fragmente. Fragmenti su dio korisničkog sučelja koji se mogu koristiti u više aktivnosti, a moguće je kombinirati i više fragmenata na jednoj aktivnosti. Ti elementi imaju svoj životni ciklus, mogu primiti vlastite ulazne podatke i obavljati svoje funkcionalnosti unutar aktivnosti. Aktivnosti pomoću fragmenata mogu upravljati s više prikaza uz korištenje puno manje resursa.



Slika 7.8. Prikaz navigacijske ladice

Prvi fragment, prikazan na slici 7.9., koji se naziva *Food_Planner_Fragment* je ujedno i najbitniji fragment aplikacije jer se na njemu odvijaju glavne interakcije s korisnikom. Na vrhu fragmenta nalazi se okvir s datumom čijim se pritiskom otvara kalendar na kojem korisnik može birati datum za koji hoće popunjavati plan prehrane. Kalendar se otvara u obliku dijaloškog okvira i promjena datuma se događa tek na pritisak „OK“ tipke. Metoda kojom se ažurira okvir s datumom također pohranjuje datum u sučelje *SharedPreferences* i osvježava popis plana prehrane i kalorijske vrijednosti metodama *LoadMealPlan* i *UpdateCalories* objašnjenih u nastavku. *SharedPreferences* se koristi za pohranjivanje malih skupina podataka u obliku parova „ključ-vrijednost“ i omogućava lako pisanje i čitanje tih podataka. U ovoj aplikaciji koristi se za pohranu datuma kako bi se pri svakom stvaranju nove instance fragmenta datum postavio na posljednji

datum koji je korisnik odabrao. Na slici 7.10. prikazana je metoda koja postavlja izabrani datum i pohranjuje ga u *SharedPreferences*.



Slika 7.9. Prikaz fragmenta *Food_Planner_Fragment*

```
private void updateLabel() {
    etDayPick.setText(sdf.format(myCalendar.getTime()));

    editor.putBoolean(MY_DATE_STATE, true);
    editor.commit();
    editor.putString(MY_DATE_VALUE, sdf.format(myCalendar.getTime()));
    editor.commit();

    adapter.stopListening();
    LoadMealPlan(String.valueOf(etDayPick.getText()));
    UpdateCalories(String.valueOf(etDayPick.getText()));
    adapter.startListening();
}
```

Slika 7.10. Metoda postavljanja datuma

Druga bitna stavka na ovom fragmentu su ukupni kalorijski unos i potrebni kalorijski unosi za svaku pojedinu kategoriju hrane koje bi korisnik za taj dan trebao ispuniti. Metoda kojom se ažuriraju ti kalorijski podaci, a koja je spomenuta ranije, je metoda *UpdateCalories*. Metoda kao

argument prihvaća datum za koji se podaci ažuriraju i preko reference na Firestore bazu podataka dohvaća vrijednosti kalorija dodane hrane i pohranjuje ih u određene varijable što je prikazano na slici 7.11. Nakon toga ta ista metoda koristi izračunati ukupni potrebni kalorijski unos korisnika, ali i potreban kalorijski unos za svaku kategoriju hrane i ažurira tekstualne okvire, odnosno *TextView* objekte.

```
private void UpdateCalories(String date) {
    TotalCalorieFromRecycler = 0;
    VegetableCalorieFromRecycler = 0;
    FruitCalorieFromRecycler = 0;
    ProteinsCalorieFromRecycler = 0;
    GrainsPastaCalorieFromRecycler = 0;

    rootRef.collection(collectionPath: "users").document(userID).collection(collectionPath: "Meal Plan").whereEqualTo(field: "Date", date)
        .get().addOnCompleteListener((task) - {
        if (task.isSuccessful()) {
            for (QueryDocumentSnapshot document : task.getResult()) {
                if (document.getString(field: "Category").equals("Fruits")) {
                    FruitCalorieFromRecycler+=document.getLong(field: "Calories").intValue();
                }

                if (document.getString(field: "Category").equals("Proteins")) {
                    ProteinsCalorieFromRecycler+=document.getLong(field: "Calories").intValue();
                }

                if (document.getString(field: "Category").equals("Grains:Pasta")) {
                    GrainsPastaCalorieFromRecycler+=document.getLong(field: "Calories").intValue();
                }

                if (document.getString(field: "Category").equals("Vegetables")) {
                    VegetableCalorieFromRecycler+=document.getLong(field: "Calories").intValue();
                }

                TotalCalorieFromRecycler+=document.getLong(field: "Calories").intValue();

                Log.d(tag: "Total Calories", msg: document.getId() + " => " + document.getData());
            }
        }
    });
}
```

Slika 7.11. Dohvaćanje kalorija iz plana prehrane u metodi *UpdateCalories*

Metode koje se koriste za računanje potrebnog kalorijskog unosa za korisnika i računanje indeksa tjelesne mase smještene su u klasi *CalculationMethods* koja u konstruktoru kao parametar prima objekt klase *User* koja sadrži korisnikove podatke potrebne za izračune. Formule korištene u tim metodama objašnjene su u četvrtom poglavlju, a dio klase *CalculationMethods* prikazan je na slici 7.12.

```

public class CalculationMethods {

    private User user;

    public CalculationMethods(User user) { this.user = user; }

    public User getUser() { return user; }

    public void setUser(User user) { this.user = user; }

    public double CalculateBMI() {
        double BMI;
        double heightInMeters = ((double) this.user.getHeight() / 100);
        BMI = this.user.getWeight() / (Math.pow(heightInMeters, 2));
        return BMI;
    }

    public int CalculateCaloriesForStagnation() {
        double BMR;
        int result;
        if (this.user.getGender().equals("Female")) {
            BMR = 655.1 + (9.563 * this.user.getWeight()) + (1.850 * this.user.getHeight()) - (4.676 * this.user.getAge());
            result = (int) (BMR * ParameterActivityLevel());
        }
        else {
            BMR = 66.47 + (13.75 * this.user.getWeight()) + (5.003 * this.user.getHeight()) - (6.755 * this.user.getAge());
            result = (int) (BMR * ParameterActivityLevel());
        }
        return result;
    }

    public int CalculateCaloriesFORWeightGain() { return CalculateCaloriesForStagnation() + 500; }
    public int CalculateCaloriesFORWeightLoss() { return CalculateCaloriesForStagnation() - 500; }
}

```

Slika 7.12. Metode klase *CalculationMethods*

Treći dio ovog fragmenta je *RecyclerView* koji prikazuje popis hrane za datum koji je trenutno odabran. Ta funkcionalnost vrši se unutar metode *LoadMealPlan* prikazane na slici 7.13. *RecyclerView* se koristi ako aplikacija treba prikazivati pomični popis objekata na temelju većih skupina podataka i podataka koji se često mijenjaju. Sastoji se od *ViewHolder*-a koji predstavlja spremnik za korisničko sučelje (sadrži objekte koji čine jedan element popisa) i adapter koji drži elemente i na temelju njih stvara određeni broj *ViewHolder*ova. Firestore baza podataka ima svoje adaptore koji u ovom slučaju dohvaćaju referencu na kolekciju plana prehrane *Meal Plan* i dohvaća sve elemente koji odgovaraju odabranom datumu.

```

private void LoadMealPlan (final String date) {

    Query query = rootRef.collection( collectionPath: "users").document(userID).collection( collectionPath: "Meal Plan").whereEqualTo( field: "Date", date);

    FirestoreRecyclerOptions<MealPlanItem> options = new FirestoreRecyclerOptions.Builder<MealPlanItem>()
        .setQuery(query, MealPlanItem.class)
        .build();

    adapter = new FirestoreRecyclerAdapter<MealPlanItem, MealPlanViewHolder>(options) {
        @NonNull
        @Override
        public MealPlanViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType) {
            View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.meal_plan_item, parent, attachToRoot: false);
            return new MealPlanViewHolder(view);
        }

        @Override
        protected void onBindViewHolder(@NonNull final MealPlanViewHolder mealPlanViewHolder, int i, @NonNull final MealPlanItem mealPlanItem) {
            mealPlanViewHolder.food_name.setText(mealPlanItem.getName());
            mealPlanViewHolder.food_quantity.setText(String.valueOf(mealPlanItem.getQuantity() + "g"));
            mealPlanViewHolder.food_category.setText(mealPlanItem.getCategory());
            mealPlanViewHolder.food_calories.setText(String.valueOf(mealPlanItem.getCalories() + "kcal"));
            mealPlanViewHolder.food_date.setText(mealPlanItem.getDate());
            mealPlanViewHolder.setItemClickListener((view, position, isLongClick) => {
                Toast.makeText(getActivity(), mealPlanItem.getName(), Toast.LENGTH_LONG).show();
            });
        }
    };

    recyclerView.setAdapter(adapter);
}

```

Slika 7.13. Metoda *LoadMealPlan*

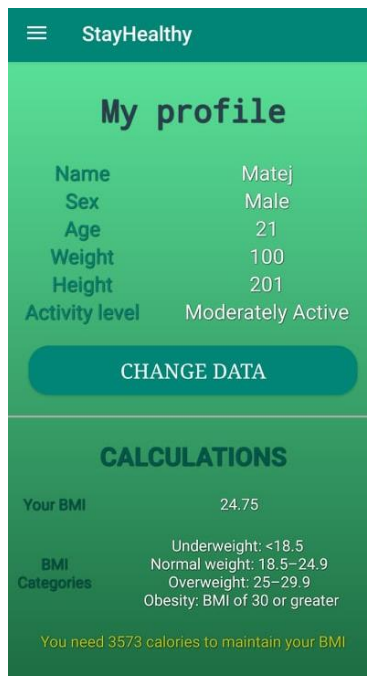
Drugi fragment aktivnosti *HomeActivity*, vidljiv na slici 7.15., je *My_Profile_Fragment* gdje korisnik ima uvid u podatke svog profila i izračune kao što su indeks tjelesne mase i potreban kalorijski unos. Osim toga na ovom fragmentu korisnik može mijenjati svoje podatke u fragmentu koji se prikazuje u obliku dijaloškog okvira, a otvara se na pritisak gumba „CHANGE DATA“. Dohvaćanje podataka korisnika preko reference Firestore baze podataka vidljivo je na slici 7.14. Podaci u dijaloškom okviru mijenjaju se na isti način kao i što se postavljaju u aktivnosti *ParameterActivity*.

```

mFirebaseAuth = FirebaseAuth.getInstance();
fStore = FirebaseFirestore.getInstance();
userID = mFirebaseAuth.getCurrentUser().getUid();
DocumentReference documentReference = fStore.collection("users").document(userID);
documentReference.get().addOnCompleteListener(new OnCompleteListener<DocumentSnapshot>() {
    @Override
    public void onComplete(@NonNull Task<DocumentSnapshot> task) {
        if(task.isSuccessful()) {
            DocumentSnapshot document = task.getResult();
            if(document != null) {
                document.getData();
                mName.setText(document.getString( field: "Name"));
                mGender.setText(document.getString( field: "Gender"));
                mAge.setText(String.valueOf(document.getLong( field: "Age")));
                mWeight.setText(String.valueOf(document.getLong( field: "Weight")));
                mHeight.setText(String.valueOf(document.getLong( field: "Height")));
                mActivityLevel.setText(document.getString( field: "Activity level"));
                user = new User(mName.getText().toString(),mGender.getText().toString(), Integer.valueOf(mAge.getText().toString()), Integer.valueOf(mWeight.getText().toString()),
                    Integer.valueOf(mHeight.getText().toString()), mActivityLevel.getText().toString());
                calculationMethods = new CalculationMethods(user);
                tvCalculatedBMI.setText(String.format("%.2f", calculationMethods.CalculateBMI()));
                UpdateCalories(tvCalorie, calculationMethods.CalculateCaloriesForOptimizingBMI(), calculationMethods.CalculateBMI());
            }
        }
    }
});

```

Slika 7.14. Programski kod dohvaćanja podataka iz Firestore baze podataka



Slika 7.15. Prikaz fragmenta *My_Profile_Fragment*

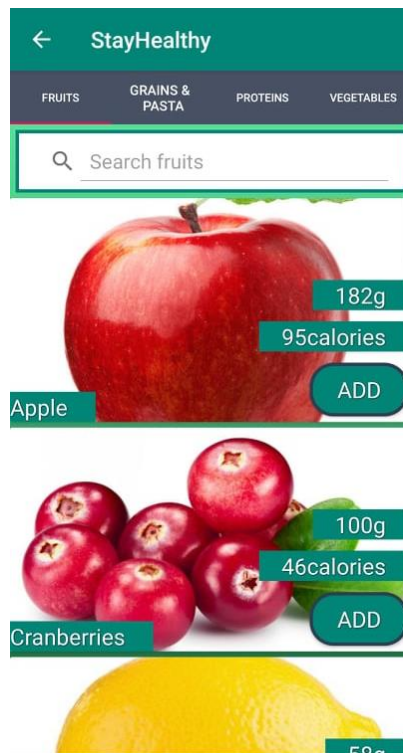
Treći fragment *Knowledge_Base_Fragment* vidljiv na slici 7.16 nema posebnu funkcionalnost osim toga što prikazuje savjete koje korisniku mogu pomoći pri vođenju zdrave prehrane.



Slika 7.16. Prikaz fragmenta *Knowledge_Base_Fragment*

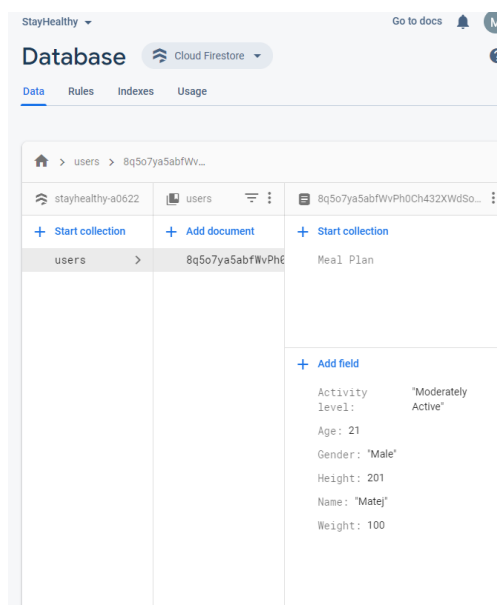
7.7. Aktivnost *FoodMenuActivity*

Ova aktivnost služi korisniku za odabir hrane iz određene kategorije koja se istog trenutka učitava u plan prehrane za postavljeni datum. Kategorije su raspoređene u četiri fragmenta koji odgovaraju svakoj pojedinoj kategoriji hrane. Prijelaz između fragmenata postignut je pomoću *ViewPager* i *TabLayout* widget-a. Svi fragmenti u sebi sadrže *RecyclerView* u kojem je prikazana hrana i opcija pretraživanja hrane po svakoj kategoriji. *FoodMenuActivity* s fragmentima raspoređenim na *TabLayout* prikazan je na slici 7.17.

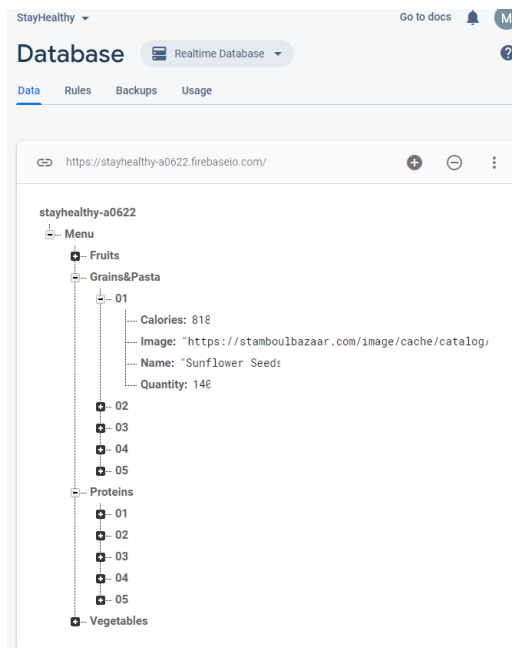


Slika 7.17. Aktivnost *Food_Menu_Activity*

Za pohranu hrane po kategorijama umjesto Firestore baze podataka korištena je Firebase baza podataka u realnom vremenu zbog jednostavnijeg učitavanja JSON (engl. *JavaScript Object Notation*) formata datoteke u kojem je pohranjena hrana. JSON je format datoteke koji koristi razumljiv tekst za prenošenje i pohranu podataka. Razlika Firestore i starije verzije Firebase baze podataka je u njihovoj strukturi što je vidljivo na slikama 7.18 i 7.19.



Slika 7.18. Firestore baza podataka



Slika 7.19. Firebase baza podataka

Metoda *UpdateRecycler*, koja je dio svakog fragmenta pojedine kategorije hrane, stvara Firebase adapter i u njega postavlja hranu pomoću klase *Food* koja je nalik ranije opisanoj klasi *User* nakon čega se adapter povezuje s *RecyclerView* objektom (Slika 7.20.).

```
private void UpdateRecycler(Query query) {
    adapter = new FirebaseRecyclerAdapter<Food, FoodViewHolder>(Food.class, R.layout.food_item, FoodViewHolder.class, query) {
        @Override
        protected void populateViewHolder(final FoodViewHolder foodViewHolder, final Food food, int i) {
            final Food local = food;
            //ADDING TO ORDER
            foodViewHolder.btnAddFood.setOnClickListener(new View.OnClickListener() {
                @Override
                public void onClick(View v) {

                    args = new Bundle();
                    args.putString("Name", food.getName());
                    args.putLong("Quantity", food.getQuantity());
                    args.putLong("Calories", food.getCalories());
                    args.putString("Category", "Fruits");
                    args.putString("Date", date);

                    ChangeFoodQuantityDialog quantityDialog = new ChangeFoodQuantityDialog();
                    quantityDialog.setArguments(args);
                    quantityDialog.show(getChildFragmentManager().beginTransaction(), tag "Change quantity dialog");
                }
            });

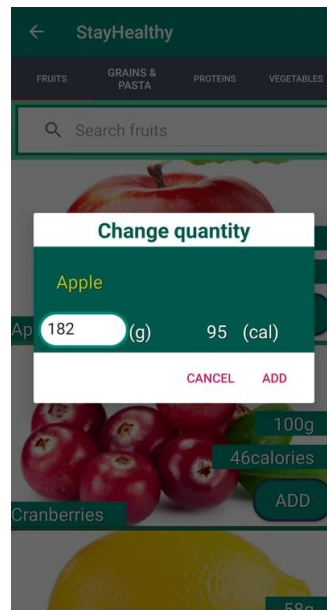
            foodViewHolder.food_quantity.setText(food.getQuantity() + "g");
            foodViewHolder.food_name.setText(food.getName());
            foodViewHolder.food_calories.setText(food.getCalories() + "kcalories");
            Picasso.with(getContext()).load(food.getImage()).into(foodViewHolder.food_image);
            foodViewHolder.setOnClickListener((view, position, isLongClick) = {
                Toast.makeText(getActivity(), local.getName(), Toast.LENGTH_SHORT).show();
            });
        }
    };

    recyclerView.setAdapter(adapter);
}
```

Slika 7.20. Metoda *UpdateRecycler*

Osim toga prethodno opisana metoda na pritisak gumba za dodavanje hrane otvara fragment u obliku dijaloškog okvira na kojem je moguće mijenjati količinu hrane koja se dodaje što je

prikazano na slici 7.21. Struktura programskog koda kojim se odabrana hrana pohranjuje u bazu podataka ima sličnu strukturu kao kod kojim se pohranjuju podaci korisnika.



Slika 7.21. Prikaz dijaloškog okvira promjene količine hrane

8. ZAKLJUČAK

Za moderno društvo tipičan je prekomjerni unos energije, loše prehrabene navike te smanjena tjelesna aktivnost i sjedilački način života. Prehrana koju čine brza hrana, neuravnoteženi obroci i smanjen ili prevelik unos hrane, dovode do slabljenja imuniteta organizma i pojave raznih bolesti, posebice kroničnih nezaraznih poput dijabetesa i pretilosti. S obzirom na veliku važnost prehrane za zdravlje i kvalitetan život ljudi, u ovom radu razvijena je aplikacija koja, prema najnovijim standardima zdrave prehrane, omogućava svakom pojedinom korisniku optimiranje i praćenje vlastite prehrane radi postizanja i održavanja poželjne tjelesne mase. Kako su Android uređaji trenutno najraširenija mobilna platforma u svijetu upravo je ona odabrana za kreiranje ove aplikacije kako bi bila dostupna što većem broju ljudi diljem svijeta.

Nakon pregleda određenih standarda zdrave prehrane kroz povijest utvrđeno je da zadnja verzija prehrabene piramide *MyPlate* iz 2011. godine najbolje zadovoljava kriterije optimirane prehrane.

Proučavanjem ostalih rješenja raspoloživih na internetu otkriveni su određeni problemi kao što je novčano plaćanje za dodatne funkcije, biranje hrane po izboru korisnika, a ne prema zdravim principima te u nekim slučajevima i prevelika složenost aplikacije. Aplikacijom *StayHealthy* nastojalo se nadopuniti navedene propuste i ukloniti nepotrebne funkcionalnosti koje korisnicima mogu otežavati rad. Uzimajući u obzir karakteristike korisnika, ova aplikacija omogućava autentikaciju te spremanje i pregled odabrane količine hrane po danima prema zdravim standardima koji su prezentirani u ovom radu. Osim tih funkcionalnosti korisnik u aplikaciji ima mogućnost mijenjati svoje podatke i optimirati personaliziranu prehranu.

Za potrebe kreiranja ove aplikacije korištena je Android Studio razvojna okolina, a baza podataka s hranom i pohrana prehrabnenih planova odvija se pomoću Firebase baze podataka. Također, za pisanje programskog koda bilo je potrebno poznavati Java programski jezik i XML označni jezik koji se koristio za uređivanje prezentacije aplikacije.

Iako je funkcionalna i u odnosu na postojeće predstavlja iskorak posebice u segmentu optimiranja zdrave prehrane prema prehrabenoj piramidi *MyPlate*, ova aplikacija ima mogućnosti za daljnje unaprjeđenje. Poboljšanje aplikacije može se postići tako da se postojeću bazu podataka proširi s dodatnom hranom za pojedine kategorije hrane. Također, vrlo jednostavno je moguće ugraditi dodatne funkcionalnosti s obzirom na to da se na glavnoj aktivnosti koristi navigacijska ladica koju se može jednostavno proširiti.

LITERATURA

- [1] »eurostat,« European Commission , 23 April 2020. [Mrežno]. Available: <https://ec.europa.eu/eurostat/en/about/overview/how-to-find-us>.
- [2] Y. C. M. K. M. T. G. S. L. B. M. Wang, »Health and economic burden of the projected obesity trends in the USA and the UK,« *The Lancet*, svez. 378, pp. 815-825, 2011.
- [3] EFSA Panel on Dietetic Products, Nutrition, and Allergies (NDA), »Scientific Opinion on establishing Food-Based Dietary Guidelines,« *EFSA Journal*, svez. 8, br. 3, p. 1460, 2010.
- [4] Škola javnog zdravstva Harvardskog sveučilišta, »Tanjur Zdrave Prehrane (Croatian) | The Nutrition Source | Harvard T.H. Chan School of Public Health,« The President and Fellows of Harvard College, [Mrežno]. Available: <https://www.hsph.harvard.edu/nutritionsource/healthy-eating-plate/translations/croatian/>. [Pokušaj pristupa 13. 07. 2020.].
- [5] C. Nishida, R. Uauy, S. Kumanyika i P. Shetty, »The Joint WHO/FAO Expert Consultation on diet, nutrition and the prevention of chronic diseases: process, product and policy implications,« *Public Health Nutrition*, svez. 7, br. 1a, pp. 245-250, 2004.
- [6] V. Katalinić, *Temeljno znanje o prehrani*, Split: Kemijsko-tehnološki fakultet Sveučilišta u Splitu, 2011.
- [7] K. M. Laquale, »Energy in-energy out: A balanced equation?,« *Athletic Therapy Today*, svez. 12, br. 5, pp. 34-37, 2007..
- [8] »Calculating Your Energy Needs | realbuzz.com,« Realbuzz Group, 22. 08. 2017.. [Mrežno]. Available: <https://www.realbuzz.com/articles-interests/nutrition/article/calculating-your-energy-needs/>. [Pokušaj pristupa 14. 07. 2020.].
- [9] »ESTIMATED ENERGY REQUIREMENT (EER) Equation - GlobalRPH,« GlobalRPH - Web Development by CI Web Design Inc., 07. 09. 2017.. [Mrežno]. Available: <https://globalrph.com/medcalcs/estimated-energy-requirement-eer-equation/>. [Pokušaj pristupa 14. 07. 2020.].
- [10] »Eat This Much - Meal Planner, Aplikacije na Google Playu,« Google, [Mrežno]. Available: <https://play.google.com/store/apps/details?id=com.eatthismuch&hl=hr>. [Pokušaj pristupa 14. 07. 2020.].

- [11] »Calorie Counter - MyNetDiary, Food Diary Tracker, Aplikacije na Google Playu,« Google, [Mrežno]. Available: <https://play.google.com/store/apps/details?id=com.fourtechnologies.mynetdiary.ad&hl=hr>. [Pokušaj pristupa 14. 07. 2020.].
- [12] »MyHealthPyramid, Aplikacije na Google Playu,« Google, [Mrežno]. Available: https://play.google.com/store/apps/details?id=com.risecreative.mypyramid&hl=hr_HR. [Pokušaj pristupa 14. 07. 2020.].
- [13] Elprocus, »What is Android? Introduction of Android OS & it's Applications,« [Mrežno]. Available: <https://www.elprocus.com/what-is-android-introduction-features-applications/>. [Pokušaj pristupa 14. 07. 2020.].
- [14] »Meet Android Studio | Android Developers,« [Mrežno]. Available: <https://developer.android.com/studio/intro>. [Pokušaj pristupa 14. 07. 2020.].
- [15] A. Dobie, »Android Studio unveiled at Google I/O keynote,« Future Us, Inc., 15. 05. 2013.. [Mrežno]. Available: <https://www.androidcentral.com/android-studio-unveiled-google-io-keynote>. [Pokušaj pristupa 14. 07. 2020.].
- [16] »What is Android SDK? - Definition from Techopedia,« Techopedia Inc. , 05. 02. 2019.. [Mrežno]. Available: <https://www.techopedia.com/definition/4220/android-sdk>. [Pokušaj pristupa 14. 07. 2020.].
- [17] S. Vrbanus, »Python prestigao Javu kao drugi najpopularniji programski jezik na GitHubu - Programiranje @ Bug.hr,« BUG MEDIA, 09. 11. 2019.. [Mrežno]. Available: <https://www.bug.hr/programiranje/python-prestigao-javu-kao-drugi-najpopularniji-programski-jezik-na-githubu-12275>. [Pokušaj pristupa 14. 07. 2020.].
- [18] »Introduction to Java,« Refsnes Data, [Mrežno]. Available: https://www.w3schools.com/java/java_intro.asp. [Pokušaj pristupa 14. 07. 2020.].
- [19] F. Lardinois, »Google Acquires Firebase To Help Developers Build Better Real-Time Apps | TechCrunch,« Verizon Media, 21. 10. 2014.. [Mrežno]. Available: <https://techcrunch.com/2014/10/21/google-acquires-firebase-to-help-developers-build-better-realtime-apps/>. [Pokušaj pristupa 14. 07. 2020.].
- [20] S. Burgess, »GIMP - A Brief (and Ancient) History of GIMP,« The GIMP Team, [Mrežno]. Available: https://www.gimp.org/about/ancient_history.html. [Pokušaj pristupa 14. 07. 2020.].

- [21] »Add build dependencies | Android Developers,« Google, [Mrežno]. Available:
<https://developer.android.com/studio/build/dependencies>. [Pokušaj pristupa 14. 07. 2020.].
- [22] »Activity | Android Developers,« Google Inc, [Mrežno]. Available:
<https://developer.android.com/reference/android/app/Activity>. [Pokušaj pristupa 14. 07. 2020.].

SAŽETAK

Jedna od najčešće korištenih prehrambenih smjernica koja je postala i sinonim za optimiranje obroka u sustavu zdrave prehrane je „prehrambena piramida“ i njezina zadnja američka verzija iz 2011. godine pod nazivom *MyPlate*. Kako bi se osigurao personalizirani pristup prehrani koji je temeljen za životnom stilu, dobi, spolu i fizičkim karakteristikama korisnika, danas se sve češće koriste različite mobilne i računalne aplikacije. U ovom radu razvijena je aplikacija koja omogućava personalizirano optimiranje prehrane s obzirom na kalorijski unos hrane te s obzirom na zastupljenost pojedinih skupina hrane u skladu s prehrambenom piramidom, odnosno s *MyPlate*.

U ovom radu razvijena je aplikacija *StayHealthy* u razvojnom okruženju Android Studio što podrazumijeva poznavanje njegovih osnovnih alata i programskog jezika Java kojim je aplikacija napisana. Za programsko rješavanje ranije navedenog problema za pohranu podataka korištena je baza podataka jedne od trenutno najpopularnijih platformi Firebase, a za potrebe uređivanja slika korištenih u aplikaciji primijenjen je uređivač slika *GIMP*.

Ključne riječi: Android, kalorijski unos, indeks tjelesne mase, plan prehrane

ABSTRACT

Android application for healthy eating planning

One of the most used nutrition guidelines that has been published and is synonymous with meal optimization of a healthy eating system is the "food pyramid", and its latest US version from 2011 is called "MyPlate". To ensure a personalized approach to nutrition that was based on lifestyle, age, gender and physical characteristics of users, mobile and computer application are used more and more. In this paper, an application has been developed that enables personalized diet optimization with caloric food intake, given the representation of certain food groups in accordance with the food pyramid, i.e. with MyPlate.

In this paper, StayHealthy was developed using Android Studio environment, which implies knowledge of the basic tools and programming languages such as Java which was used to write the application. Firebase database, which is a part of one of the most popular platforms, was used as a solution for storage requirements, and GIMP editor was used for image editing purposes.

Keywords: Android, caloric intake, body mass index, diet plan

ŽIVOTOPIS

Matej Kovačević je rođen 11.3.1999. godine u Zagrebu. Pohađao je Osnovnu školu Retfala u Osijeku. Tijekom pohađanja sudjelovao je na školskim natjecanjima iz matematike te županijskom natjecanju iz kemije. Nakon završene osnovne škole, 2013. godine upisuje se u III. Gimnaziju Osijek. Nakon završetka gimnazije, upisuje se na preddiplomski sveučilišni studij Računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Koristi se programskim jezicima C++, C# i Java te se služi razvojnim okruženjem Androida.

POPIS SLIKA

Slika 3.1. Tanjur zdrave hrane "MyPlate", izvor: [4].....	5
Slika 3.2. Prehrambena piramida prema Školi javnog zdravlja T. H. Chan harvardskog sveučilišta, izvor: [4].....	7
Slika 5.1. Dijagram funkcija aplikacije.....	16
Slika 6.1: Prikaz dijelova Android OS-a, izvor: [13].....	18
Slika 6.2. Razvojno okruženje Android Studio.....	19
Slika 6.3. XML datoteka.....	21
Slika 6.4. Uređivanje glavne slike aplikacije <i>StayHealthy</i>	23
Slika 7.1. Gradle datoteka.....	24
Slika 7.2. Logotip aplikacije.....	25
Slika 7.3. Aktivnost <i>MainActivity</i>	26
Slika 7.4. Registracija korisnika u aktivnosti <i>SignUp</i>	27
Slika 7.5. Aktivnost <i>ParameterActivity</i>	27
Slika 7.6. Prikaz klase <i>User</i>	28
Slika 7.7. Pohrana podataka korisnika u bazu podataka.....	28
Slika 7.8. Prikaz navigacijske ladice.....	29
Slika 7.9. Prikaz fragmenta <i>Food_Planner_Fragment</i>	30
Slika 7.10. Metoda postavljanja datuma.....	30
Slika 7.11. Dohvaćanje kalorija iz plana prehrane u metodi <i>UpdateCalories</i>	31
Slika 7.12. Metode klase <i>CalculationMethods</i>	32
Slika 7.13. Metoda <i>LoadMealPlan</i>	32
Slika 7.14. Programski kod dohvaćanja podataka iz Firestore baze podataka.....	33
Slika 7.15. Prikaz fragmenta <i>My_Profile_Fragment</i>	33
Slika 7.16. Prikaz fragmenta <i>Knowledge_Base_Fragment</i>	34
Slika 7.17. Aktivnost <i>Food_Menu_Activity</i>	35
Slika 7.18. Firestore baza podataka.....	35
Slika 7.19. Firebase baza podataka.....	36
Slika 7.20. Metoda <i>UpdateRecycler</i>	36
Slika 7.21. Prikaz dijaložkog okvira promjene količine hrane.....	37

POPIS TABLICA

Tablica 3.1. Prehrambene preporuke WHO i FAO, izvor: [5].....	6
Tablica 4.1. Koeficijenti tjelesnog napora prema dobnim skupinama, spolu i	12

PRILOZI

Na priloženom DVD disku su sadržani:

- Android Studio projekt
- Završni rad u .pdf i .docx formatu