

MOBILNA ANDROID APLIKACIJA ZA POTPORU KONTROLIRANOJ I ZDRAVOJ PREHRANI

Barbarić, Nikola

Undergraduate thesis / Završni rad

2019

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:200:094030>

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

Download date / Datum preuzimanja: **2024-11-24**

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

MOBILNA ANDROID APLIKACIJA ZA POTPORU

KONTROLIRANOJ I ZDRAVOJ PREHRANI

Završni rad

Nikola Barbarić

Osijek, 2019.

SADRŽAJ

1. UVOD	1
1.1. Zadatak završnog rada	2
2. IZAZOVI I PROBLEMI VEZANI ZA ZDRAVU PREHRANU.....	3
2.1. Zdrava prehrana	3
2.1.1. Definicija zdrave prehrane	3
2.1.2. Zablude zdrave prehrane	4
2.2. Problemi nezdrave prehrane i potreba za nadzorom prehrane	4
2.3. Modeliranje zdrave prehrane	4
2.3.1. B. J. Devine-ova izraz	4
2.3.2. Kalorije.....	5
2.3.3. Primjena modela zdrave prehrane u realnom slučaju.....	6
3. RAČUNALNE TEHNOLOGIJE U KONTROLI PREHRANE.....	8
3.1. Usporedba gotovih rješenja na različitim platformama.....	8
3.1.1. Calorie Counter – MyFitnessPal	8
3.1.2. Lose It!	9
3.1.3. NutriMenza.....	10
3.2. Idejno rješenje mobilne aplikacije za kontrolu prehrane.....	11
4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE	13
4.1. Android okolina.....	13
4.1.1. Općenito o Androidu.....	13
4.1.2. Arhitektura i inačice Androida.....	14
4.2. Razvojne okoline, tehnologije i jezici	15
4.2.1. Android Studio	15
4.2.2. Java.....	16

4.2.3.	Opisni jezik XML.....	17
4.2.4.	Grafički program Photoshop	18
4.2.5.	Firebase	19
4.3.	Programske komponente i razvoj mobilne aplikacije.....	19
4.3.1.	Android Manifest	19
4.3.2.	Aktivnosti	20
4.3.3.	SharedPreferences	24
4.3.4.	RecyclerView	28
4.3.5.	Fragmenti	28
4.3.6.	FireBase.....	29
5.	PRIKAZ KORIŠTENJA I ISPITIVANJE APLIKACIJE SA ANALIZOM REZULTATA.....	31
5.1.	Ispitivanje funkcionalnosti aplikacije.....	31
5.1.1.	Prvo ispitivanje, slučaj kalorijskog deficita	31
5.1.2.	Drugo ispitivanje, slučaj kalorijskog suficita	32
5.1.3.	Treći slučaj, kalorijska nula	33
5.2.	Usporedba rezultata	33
6.	ZAKLJUČAK	35
	LITERATURA.....	36
	ŽIVOTOPIS	38
	SAŽETAK.....	39
	ABSTRACT	40
	PRILOZI.....	41

1. UVOD

Unatoč globalnoj dostupnosti interneta, te velikom broju online savjeta za mršavljenje (debljanje), većina ljudi, a i studenata bori se s viškom kilograma. Kako navodi Hrvatski zavod za javno zdravstvo[7] problemi su epidemijskih razmjera. Više od 700 milijuna ljudi u svijetu ima problem s debljinom. Kada je časopis Clin Dermatol [9] prikazao rezultate u Sjedinjenim Američkim Državama, došlo se do rezultata da 65% stanovništva pati od prekomjerne tjelesne mase, dok 31% odraslih osoba ima debljinu. Na području Europe više je od 40% odraslih osoba s prekomjernom tjelesnom masom, a 20% s debljinom. U Hrvatskoj s debljinom je 20,37% odraslog stanovništva. Debelo je 20,59% žena i 20,13% muškaraca, od čega s godinama između 45 i 54 nalazi se 27,86% muškaraca i 32,83% žena. Studenti kao ciljna grupa ove aplikacije su više tehnološki osviješteni, ali prema zadnjim istraživanjima dolazi se do podatka da sve više mladih nije zadovoljno svojom težinom. Aplikacija prati pristup i rješavanje problema nezdrave prehrane, odnosno kontrolirane ishrane.

Aplikacija je bazirana na ideji, da korisniku nakon što unese svoje podatke, prati i evidentira ishranu, broj unesenih kalorija, te prikazuje njegov napredak do idealne težine. Budući da su ciljna grupa studenti, aplikacija ima kompletnu ponudu glavnih jela i priloga svih studentskih menzi, te na taj način olakšava izbor jela. Aplikacija će se izrađivati u Android Studio, te će ju biti moguće pokrenuti na skoro svim uređajima koji pokreću Android operacijski sustav. Za razumijevanje koda ove aplikacije potrebno je poznavanje programskog jezika Java, ali i poznavanje rada programskog okruženja Android Studio-a. Kao jedan od najčešćih izbora za bazu podataka izabran je Firebase, a za obradu slika program Photoshop.

U drugom poglavlju završnog rada prikazani su izazovi i problemi vezani za zdravu prehranu, njena definicija, kao i najčešće zablude vezane za nju. Modelom zdrave prehrane objašnjavaju se izrazi potrebni za izračun idealne mase, te preporučenog dnevnog kalorijskog unosa. Treće poglavlje objašnjava računalne tehnologije u kontroli prehrane, prikazuje usporedbu gotovih rješenja na različitim platformama, te donosi idejno rješenje aplikacije prema kojemu je i napravljena. Četvrto poglavlje pokazuje i objašnjava Android okolinu, razvojne tehnologije koje su korištene prilikom izrade, te prikazuje najvažnije komponente koje se upotrebljavaju u projektu. U petom poglavlju aplikaciju se podvrgava ispitivanjima da bi se provjerila njena ispravnost i funkcionalnost.

1.1.Zadatak završnog rada

U teorijskom dijelu rada treba razraditi plan i pojedinosti za kontroliranu i zdravu prehranu za osobe s normalnom, prekomjernom i nedovoljnom tjelesnom težinom u ovisnosti o osobinama korisnika, njegovom načinu života ili postojećoj bolesti. Također, treba razraditi model i predložak arhitekture mobilne aplikacije s bazom podataka. Aplikacija treba omogućiti registriranje i prijavu korisnika, unos, pohranu i analizu podataka o osobinama, načinu života i ciljevima prehrane, prikaz dnevnog unosa, unosa u duljem razdoblju, te generiranje preporuka i upozorenje korisniku. Pri tome treba po potrebi koristiti odgovarajući postupak analize podataka. U praktičnom dijelu rada mobilnu Android aplikaciju treba programski ostvariti, ispitati na odgovarajućem skupu ulaznih podataka i prikladno analizirati.

2. IZAZOVI I PROBLEMI VEZANI ZA ZDRAVU PREHRANU

Unutar ovoga dijela završnog rada bit će objašnjena problematika ljudskog organizma, definicija, problemi i zablude zdrave prehrane. Model zdrave prehrane prikazuje osnovne izraze na kojima se zasniva izračun dnevnog unosa kalorija i idealne mase.

2.1. Zdrava prehrana

Iako zdrava prehrana najviše podsjeća na mršavljenje, ona se najprije odnosi na zdravlje. Loše stanje tijela uzrokovano nutricionistički neadekvatnom prehranom dovodi do posljedica poput poremećaja težine. Dovodjenje tijela u red podrazumijeva opskrbu vitaminima, proteinima, masnim kiselinama, mineralima, aminokiselinama, ugljikohidratima... svime potrebnim za obavljanje normalne funkcije. Međutim, između svih tih sastojaka koji čine prehranu, sakrivaju se oni koji su pozitivni za tijelo, ali i oni negativni. Bitno je znati razvrstati loše od dobrih. Na naše tijelo osim vrste hrane utječe i količina koja se unosi. Kako navode Važan dio zdrave prehrane je doziranje. Kako manjak može biti negativan, negativan može biti i višak. Nutricionisti imaju razne preporuke od kontrole količine obroka i njihovog broja do praćenja vremena konzumacije. Ipak, kako navodi Kadulja [8] najvažnije kod obroka je nutricionistička vrijednost koju on posjeduje.

2.1.1. Definicija zdrave prehrane

Internet je prepun raznih teorija o zdravoj prehrani, međutim bit će uvrštena definicija sa sajma svjetske zdravstvene organizacije[12]. Pod zdravom prehranom podrazumijeva se konzumiranje namirnica bogatih nutricionističkim vrijednostima iz svake skupine hrane i to u količinama koje su preporučene.

Hrana koja se konzumira mora zadovoljavati sljedeće uvjete da bi se smatrala zdravom:

- namirnice moraju imati pozitivan učinak na zdravlje
- obroci u preporučenim količinama koje definira piramida zdrave prehrane
- adekvatna priprema namirnica zbog očuvanja nutritivnih vrijednosti
- raspored obroka kroz dan
- obroci u približno isto vrijeme
- jesti pravilnim načinom – hrana mora biti dobro prožvakana, ne smije biti prebrzo pojedena
- mora biti nepokvarena, te skladištena preporučeno

2.1.2. Zablude zdrave prehrane

Zdravo se hraniti ne podrazumijeva lišavanje obroka i namirnica, stalno gladovanje i jedenje samoga povrća. Važno je izbalansirati sve oblike namirnica i od svake uzeti dio koji je tijelu potreban. Vođeno tim pravilom, vidljivo je da su tijelu najviše potrebni vitamini i minerali koji se uzimaju iz voća i povrća, zato baš ta ishrana treba biti najzastupljenija u dnevnim obrocima. Međutim to nije jedino što je potreba. Tijelo treba i ugljikohidrate i masti, koji su često smatrani nezdravim. Bitno je znati izabrati zdrave inačice ovih namirnica, tj. one koje će tijelu donijeti više koristi nego štete.

2.2. Problemi nezdrave prehrane i potreba za nadzorom prehrane

Jedan od glavnih problema koji muči svijetu koji žele smršaviti je neinformiranost, te nedovoljno poznavanje energetske vrijednosti pojedinih sastojaka koji se svakodnevno jedu. Jedno od istraživanja provedeno u Velikoj Britaniji pokazuje da se puno veći broj ispitanika pridržava nečega ako ima zadani plan i program u odnosu na one kojima je ostavljeno da sami istražuju. To je cilj aplikacije da omogući korisniku jednostavno sučelje za evidenciju, a dok se računanja i evidencija odvijaju u pozadini, bez potrebe korisnika za samostalnim analizama.

2.3. Modeliranje zdrave prehrane

U ovom dijelu završnoga rada, kroz kratki prikaz najvažnijih čimbenika koji utječu na zdravu prehranu, bit će prikazan model zdrave prehrane. Kroz dugi niz godina sljedeći izrazi i izračuni su ispitivani, provjeravani, te su se pokazali u većini slučajeva kao točni. U odstupanje se naravno moraju uzeti ljudi koji pate od neke bolesti, pa se njihov organizam drugačije ponaša, no oni nisu ciljna skupina ove aplikacije, jer su u manjini i njihov izračun je mnogo složeniji, te zahtjeva stalnu kontrolu liječnika.

2.3.1. B. J. Devine-ova izraz

Korisniku je jedino zanimljivo da vidi svoju idealnu masu, a ne zanima ga kako se ona računa u pozadini, no za njeno računanje potreban je izraz kojeg je još 1964. godine metodom brojnih uzoraka izveo doktor G. J. Hamwi. Deset godina kasnije kako navode u literaturi[6], formulu je malo prepravio B. J. Devine te je s vremenom ovaj izraz postao univerzalnim za računanje idealne tjelesne mase. Originalni izraz je u inčima, međutim ovdje će biti prikazana u centimetrima.

Muškarci: 50.0 kg + 2.3 kg za svakih 2.54 centimetra preko 152.4 centimetra

(2-1)

Žene: 45.5 kg + 2.3 kg za svakih 2.54 centimetra preko 152.4 centimetra

2.3.2. Kalorije

Kalorija je mjerna jedinica i u prehrani se koristi za mjerenje energetskog sastava neke hrane. Kalorija u hrani podrazumijeva kilokalorija ili 1000 kalorija, ali se u govoru uobičajila i koristi samo kalorija umjesto kilokalorija. Tako kada se kaže kako proteini imaju 4 kalorije po gramu, zapravo se misli na 4 kilokalorije po gramu, tako da će se u nastavku teksta i govoriti, a i označavati s kcal. Svaka namirnica se sastoji od masti, ugljikohidrata ili proteina, ili kombinacija ova tri sastojka. Masti imaju 9 kalorija po gramu, proteini 4, a ugljikohidrati također 4 kalorije po gramu. Kroz aplikaciju će se pratiti i unos pojedinih osnovnih elemenata, jer su npr. za mišićnu masu potrebni proteini nego ugljikohidrati.

2.3.2.1. Dnevni unos kalorija

Koliko je preporučeni dnevni unos kalorija ovisi o veličini tijela, ali i o razini dnevne aktivnosti. Krupnija osoba treba više kalorija od mršavije osobe za održavanje određene tjelesne mase, ali postoje neki osnovni proračuni, pomoću kojih se može saznati koliko prosječna osoba treba kalorija uz prosječnu razinu aktivnosti. Ukoliko je netko teži i kreće se više bit će mu potrebno nešto više kalorija, a dok će netko tko vodi sjedilački životni režim trebati manje kalorija nego prosječna osoba. Žena prosječne tjelesne mase i s prosječnom aktivnosti ima dnevnu potrošnju kalorija oko 2000 do 2500 kcal. Muškarac prosječne tjelesne mase i prosječne aktivnosti ima dnevnu potrošnju kalorija od 2500 do 3000. Osobe koje su starije od 40 godina trebaju smanjivati kaloričnost hrane u prosjeku za svaku godinu po jedan posto od dnevne doze unosa kalorija.

2.3.2.2. Kalorijski deficit / suficit

Jedan kilogram masti u tijelu ima oko 7700 kalorija, tako da jedan kilogram će osoba smršaviti kada unese 7700 kalorija manje. Npr. ukoliko je dnevna potrošnja 2800 kalorija, smanji li se unos za 500 kalorija dnevno, bit će potrebno 14 dana da se smršavi jedan kilogram, kao što pokazuje Nikad ne bi trebalo smanjivati dnevni unos kalorija ispod 1500 kcal, jer tada se događa usporavanje metabolizma, pa će biti potrebno duže vremena za smršaviti, a brojni stručnjaci kažu da je najoptimalnije 500

kalorija dnevno biti u deficitu, odnosno suficitu, te zbog toga će ta brojka biti korištena u aplikaciji za proračune jer je to način zdravog mršavljenja.

2.3.2.3. Harrs Benedict-ova izraz

Tijelo dnevno troši određeni broj kalorija, te za mršavljenje je bitno biti ispod toga broja kalorija s dnevnim unosom kalorija. Koliki je taj broj, zavisi od više stavki, kao što su: visina, težina, spol, godine, aktivnost. To su osnovne stvari koje najviše utječu na taj broj, ali između ostalog svoj utjecaj imaju i genetika, bolesti, rad srca, stres. Ljudsko tijelo je poput stroja kojemu je svaki dan potrebna energija za rad, održavanje, izgradnju odnosno razgradnju različitih tvari. Osim navedenog potrebna mu je i energija za sve ostale aktivnosti koje mora obavljati. [6] Najčešće korišten izraz za izračun kalorija koje su potrebne za održavanje trenutne tjelesne kilaže je izraz znanstvenika Harrs-Benedict-a koji glasi ovako:

Izraz za muškarce:

$$88.362 + (13.397 \times \text{tjelesna masa u kg}) + (4.799 \times \text{visina u cm}) - (5.677 \times \text{godine}) \quad (2-2)$$

Izraz za žene:

$$447.593 + (9.247 \times \text{tjelesna masa u kg}) + (3.098 \times \text{visina u cm}) - (4.330 \times \text{godine}) \quad (2-3)$$

Nadalje izračun je potrebno pomnožiti sa koeficijentom aktivnosti da bi se dobio ukupni broj dnevnih kalorija koje su potrebne:

Sjedilački način života bez aktivnosti:	broj kalorija x 1.2	
Mala dodatna aktivnost:	broj kalorija x 1.375	
Umjerene aktivnosto:	broj kalorija x 1.55	(2-4)
Velike količine aktivnosti:	broj kalorija x 1.725	
Jako velike količine aktivnosti:	broj kalorija x 1.9	

2.3.3. Primjena modela zdrave prehrane u realnom slučaju

Za objašnjenje principa zdravog kalorijskog deficita za one koji žele smršaviti odnosno suficita za slučaj gdje se potrebno udebljati kao primjer bit će predstavljen student Luka, osoba od 20 godina, visine 182 cm, te s kilažom od 85 kilograma. Luka je prosječni student bez neke veće fizičke aktivnosti

tijekom tjedna, ali je odlučio da smršavi. Uvrštavajući svoju visinu u izraz (2-1), Luka dobiva da je njegova idealna težina 77 kilograma. Nakon izračuna razlike u kilogramima, odnosno mase koju Luka mora smršaviti, računa se vrijeme potrebno za taj pothvat. Za njegov primjer vidljivo je kako mora smršaviti $85-77=8$ kilograma. Znajući da je jedan kilogram jednak 7700 kalorija, vidimo da mora izgubiti 61600 kalorija, a ako svaki dan gubi 500 kalorija, bit će mu potrebno oko 5 mjeseci. Ova metoda je sigurno učinkovita i ne vraćaju se kilogrami nakon nje, zato i ne obećava brze rezultate. Mnoge dijete širom interneta garantiraju gubitak 8 kilograma u bržem vremenskom roku, ali one ili podrazumijevaju izgladnjivanje, ili je to samo gubitak vode iz organizma, koji se vrlo lako vrati. Ovom metodom na kojoj je bazirana aplikacija se tope masne naslage, te provjereno djeluje, ali iziskuje vrijeme.

3. RAČUNALNE TEHNOLOGIJE U KONTROLI PREHRANE

Sljedeći dio završnoga rada prikazat će usporedbu gotovih rješenja za zdravu i reguliranu prehranu na različitim platformama, navesti njihove glavne prednosti i nedostatke, te izgled korisničkog sučelja. Bit će prikazano i idejno rješenje ove aplikacije, prema kojemu je i rađena u nastavku.

3.1. Usporedba gotovih rješenja na različitim platformama

Na internetu postoji mnogo rješenja koja se bave dnevnim unosom kalorija, kontroliranjem prehrane, te praćenjem i savjetovanjem oko težine, aktivnosti i ishrane. Bit će izdvojeno par najpopularnijih, a to su: Calorie Counter – MyFitnessPal, Lose It!, NutriMenza. Detaljnom analizom tih aplikacija, utvrđeno je da svaka posjeduje određene prednosti, kao i mane. Riječ je samo o Android aplikacijama, međutim postoji određeni broj aplikacija za stolna računala, koje bi mogle zauzeti mjesto na rang listi. Svi podatci o aplikacijama preuzeti su sa službene Google Trgovine Play stranice.

3.1.1. Calorie Counter – MyFitnessPal

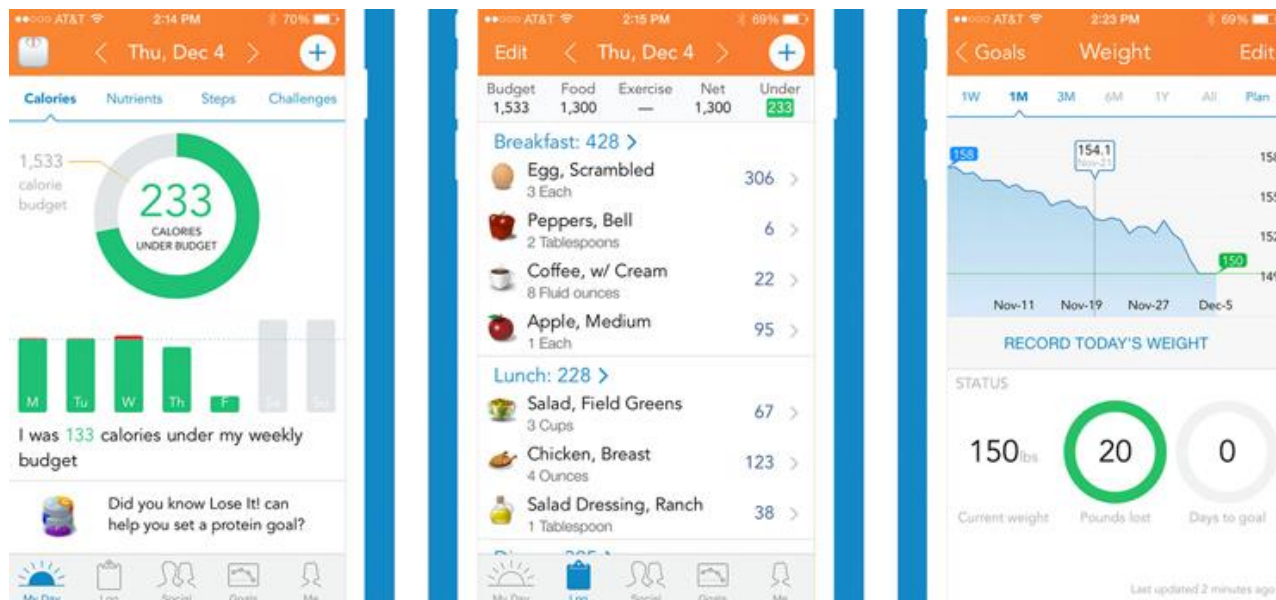
MyFitnessPal je svjetski poznata aplikacija, dostupna na Android i iOS uređajima koja broji preko 50 milijuna preuzimanja samo na Android uređajima. Rađena je u suradnji s brojnim nutricionistima, te fitness trenerima, a na slici 3.1 vidi se izgled korisničkog sučelja, a dostupna za preuzimanje je putem Trgovine Play [11]. Preporučuje se naprednijim korisnicima koji žele pratiti svaki svoj dnevni korak, te se kao takva može smatrati i pozitivnom i negativnom, ovisno od toga koliko je napredan korisnik. Međutim prosječan korisnik ne želi previše vremena gubiti u aplikaciji, te rješavati brojne upitnike. Prednosti aplikacije su velika baza podataka sa svjetskim jelima, te mogućnost skeniranja crtičnih kodova na proizvodima i automatsko očitavanje kalorijskih vrijednosti. Međutim kao nedostatak joj se navodi nedostatak jela specifičnih za hrvatsko područje i obaveza davanja osobnih podataka, zbog afere 2018. godine [5] kada je 144 milijuna elektoničkih adresa, zajedno sa korisničkim imenima, lozinkama i IP adresama procurilo, da bi se kasnije iste pronašle u prodaji na crnom internetu.



Slika 3.1 Korisničko sučelje aplikacije MyFitnessPal

3.1.2. Lose It!

Lose It! je aplikacija za brojanje kalorija koja pomaže u cilju postizanja željene težine. Od izlaska 2008. godine istaknuta je u brojnim novinama, raznim portalima te je česta preporuka i fitness trenera. Prednosti aplikacije su velika baza podataka jela kao što se navodi na stranici [10], mogućnost postavljanja izazova koje je potrebno savladati, te povezivost s pametnim narukvicama i satovima. Nedostatak joj je potreba za internetom, količina obavijesti koje šalje, ali i nedostatak Hrvatskoj specifičnih jela kao i kod prethodno navedene aplikacije. Na slici 3.2 prikazan je izgled određenih aktivnosti aplikacije Lose It.



Slika 3.2 Izgled aplikacije Lose It!

3.1.3. NutriMenza

Ova se aplikacija našla na popisu, iako nije popularna na svjetskoj razini, zbog toga što je hrvatski proizvod, te kao takva ima određene prednosti. Pozitivnu ocjenu ovoj aplikaciji, kako navodi Studentski[9] daje pokrivenost svih jela iz menze, te je na taj način studentima koji su ciljna grupa i ovoga završnoga rada, olakšan izbor. Nedostatak aplikacije je potreba za stalnom internet vezom, te nemogućnost spremanja napretka na kraju dana. [9] Na slici 3.3 je prikazan izgled jednoga od zaslona.

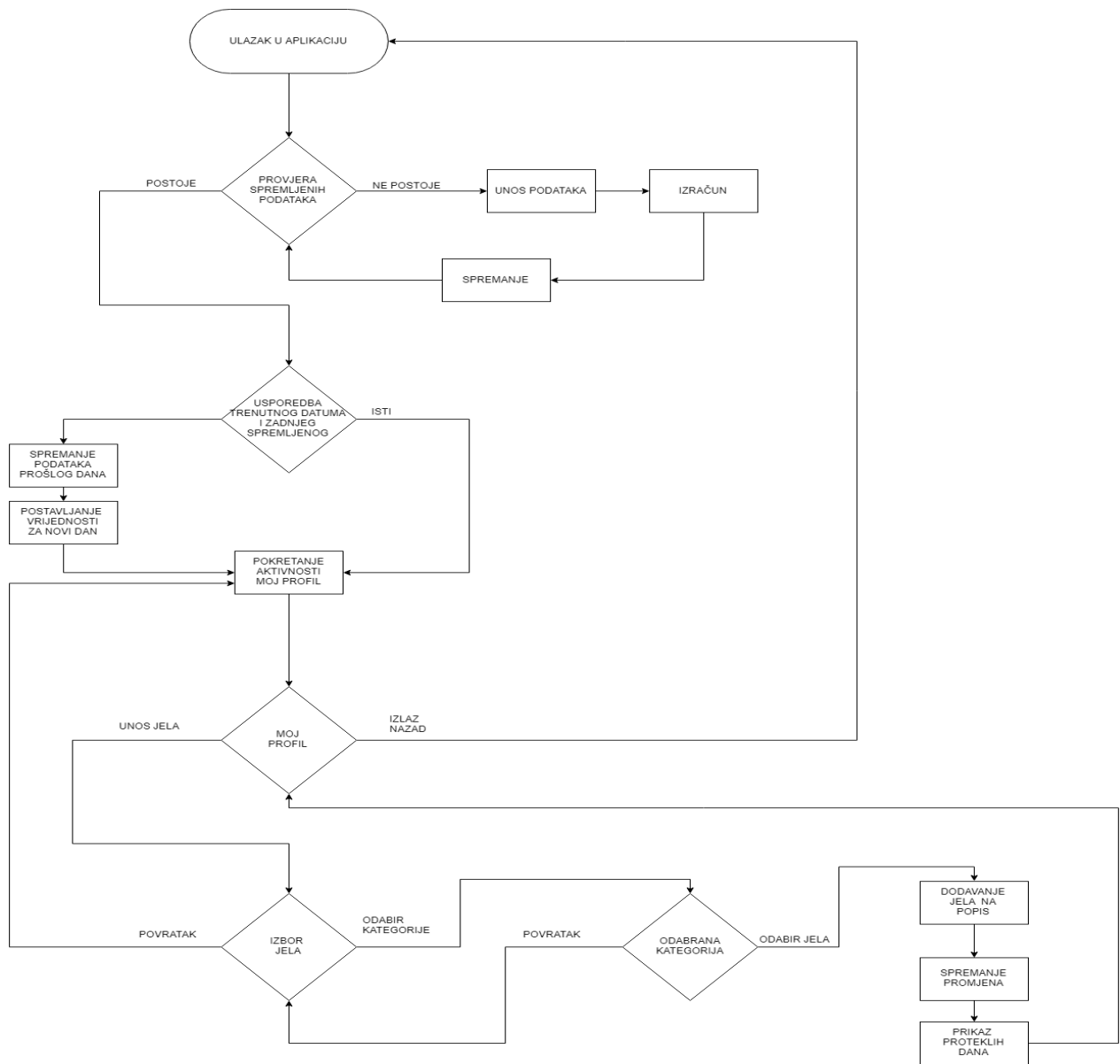


Slika 3.3 Prikaz aplikacije NutriMenza

3.2. Idejno rješenje mobilne aplikacije za kontrolu prehrane

Prije izrade same aplikacije potrebno je osmisliti i temeljito proučiti idejno rješenje, kojega se kasnije treba pridržavati. U obzir treba uzeti razne zahtjeve koje aplikacija mora poštivati.

Prilikom ulaska u aplikaciju obaviti će se provjera da li postoje prethodno spremljeni podatci. Na taj način je postignuto da korisnik samo jednom unosi svoje podatke, a da svaki sljedeći put ih aplikacija zna. Ukoliko nema spremljenih podataka, omogućit će se unos podataka, na osnovu kojih će se obaviti izračun prema dijelu opisanom u prethodnom poglavlju. Dobivene vrijednosti će biti spremljene, te učitavane prilikom svakog sljedećeg pokretanja. Osim korisnikovih podataka spremaju se još neki podatci poput trenutnog datuma, pomoću kojega se provjerava da li je u pitanju novi dan. Budući da se radi o dnevnom unosu kalorija potrebno je prilikom svakoga pokretanja aplikacije provjeriti da li je nastupio novi dan, te ako jest postaviti početne vrijednosti za novi dan, a vrijednosti prethodnoga dana spremati te ih s proteklim danima prikazati u stupčanom grafikonu. Korisniku se prikazuju njegove osnovne informacije, poput mase, idealne mase, kao i u kalorijama izraženo koliko treba smršaviti ili udebljati se, itd. Klikom na jelovnik, otvaraju mu se kategorije jela, gdje daljnjim odabirom ulazi u određenu potkategoriju. Unutar te potkategorije, prikazana su sva jela iz te potkategorije, na primjer, glavna jela. Odabirom određenog jela ono se dodaje na novi popis jela toga dana. Korisnik se može kretati kroz kategorije, te dodavati jela koliko želi, a sve promjene se automatski spremaju, te su spremljene za sljedeće pokretanje aplikacije.



Slika 3.4 Dijagram prikaza funkcionalnosti aplikacije

4. PROGRAMSKO RJEŠENJE MOBILNE APLIKACIJE

Tehnologije i principi koji se koriste za izradu ove aplikacije bit će opisani u nastavku, te će biti navedene osnove o Androidu, razvojnim okolinama, tehnologijama, kao i jezicima. Detaljnije će biti opisani programske komponente koje su korištene za razvoj mobilne aplikacije.

4.1. Android okolina

Aplikacija je prvotno namijenjena za Android uređaje, te će u nastavku biti opisan Android te njegova arhitektura, kao i inačice. Razvojne tehnologije i programi korišteni prilikom izrade su također objašnjeni u ovom dijelu. Znati teorijsku podlogu o Android okolini je bitno, radi različitih smjerova u kojima je moguće razvijati Android aplikaciju.

4.1.1. Općenito o Androidu

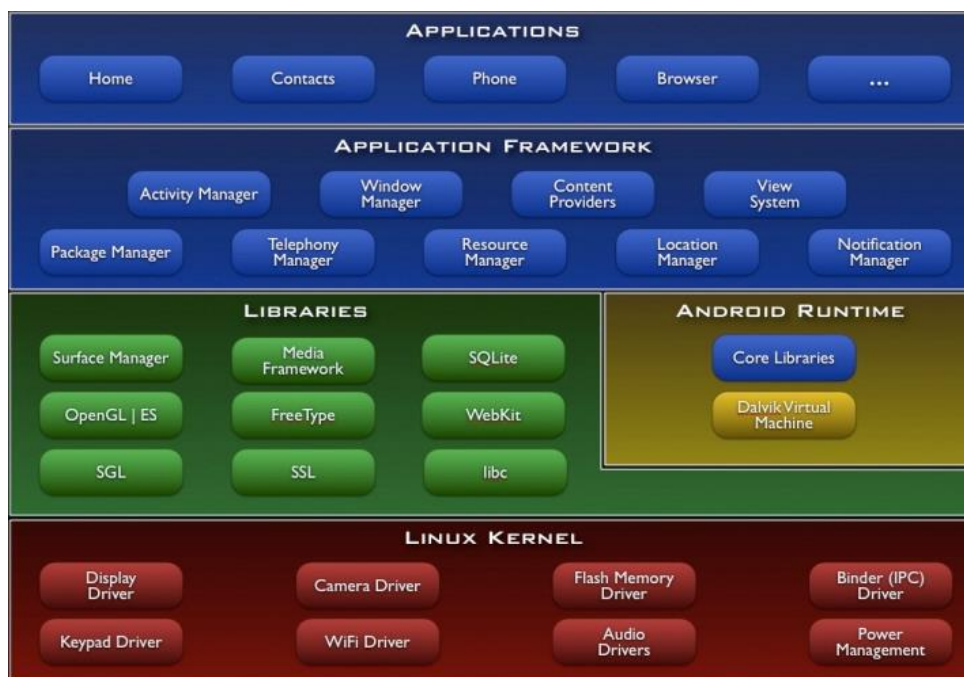
Android je operacijski sustav pisan otvorenim kodom za mobilne uređaje, pametne telefone i tablet računala, tvrtke Google iz Amerike. Temelji se na jezgri Linux-a i ostaloj programskoj podršci otvorenog koda. Zatim, Google razvija i Android TV za pametne televizore, Android Auto za automobile i Wear OS na pametnim satovima. Isprva je Android razvijan u tvrtki Android Inc. dok ih nije 2005. godine kupio Google. Prvi je puta predstavljen u javnosti 2007. godine, dok su se prvi uređaji koji se temelje na Androidu pojavili u devetom mjesecu 2008. godine. Pod licencom Apache je dostupan temeljni programski kôd Androida poznat i kao Android projekat otvorenog koda (AOSP). Android je vezan s paketom vlasničke programske podrške razvijene od strane Google-a, što uključuje servise poput Gmail-a ili tražilice Google, trgovinu s aplikacijama ili platformu za prodaju digitalnog sadržaja Google Play. Proizvođači koji svoje uređaje temelje na Androidu moraju licencirati tu programsku podršku prema uvjetima Google-a. Ostali operacijski sustavi koji se temelje na AOSP-u, poput Amazon-ovog Fire OS-a, moraju koristiti sopstvene servise umjesto Google-ovih.

Trenutno je Android mobilni sustav za pametne telefone sa najviše prodanih modela još od 2011. kao i za tablete još od 2013. godine. Prema podacima[3] iz svibnja 2019. godine, Android broji više od 2.5 milijarde korisnika koji su aktivni na mjesečnom nivou, daleko više od bilo kojeg od ostalih operacijskih sustava, dok na svjetskoj razini Android zauzima 85% tržišta.

4.1.2. Arhitektura i inačice Androida

Operacijski sustav Android je sastavljen od više dijelova i slojeva. Svaki sloj ima određene karakteristike i ulogu, ali slojevi nisu baš najjasnije odvojeni već se često preklapaju jedni s drugima. Baziran je na Linux jezgri 2.6 koja je napisana u C ili C++-u.

Linux je operacijski sustav s otvorenim kodom, pa aplikacije mogu preko posrednika komunicirati ili pokretati druge aplikacije poput slanja SMS poruka ili elektroničkih pošti, fotografiranja, primanja poziva i ostalog. Razlog zbog kojega se koristi su njegova laka prenosivost, sigurnost te značajke. Budući da se sustavi koji se temelje na Linuxu ne moraju brinuti o sklopovskim karakteristikama pojedinih uređaja time je omogućeno da se implementira Android na relativno veliki broj uređaja. Za radno okruženje su korišteni C i C++, ali većinom se kod piše u Javi uz Android SDK. Pisati je moguće i u C odnosno C++, ali se tada je potrebno koristiti Android NDK. Ovim postupkom omogućeno je bolje raspolaganje resursima uređaja ili korištenje knjižnica iz jezgre i radnog okruženja a sama brzina aplikacije može se povećati i do deset puta. Mana ovakvog pristupa je što zahtjeva kompleksnije pisanje koda. Arhitektura Android operacijskog sustava prikazana je na slici ispod.



Slika 4.1 Arhitektura Android operacijskog sustava

Na slici 4.2 vidljive su inačice Androida poredane redosljedom kojim su izlazile, te njihovi nazivi. Nisu nabrojane inačice prije 4.0.x, jer se te inačice više ne koriste.

4.0.x	Ice Cream Sandwich	19. listopada 2011.	14-15
4.1/4.2/4.3	Jelly Bean	9. srpnja 2012., 4.2.1 - 27. studenoga 2012.	16-18
4.4	KitKat	3. rujna 2013.	19-20
5.0/5.1	Lollipop	25. lipnja 2014.	21-22
6.0 / 6.0.1	Marshmallow	5. listopada 2015. ^[1]	23
7.0/7.1	Nougat	22. kolovoza 2016.	24-25
8.0/8.1	Oreo	21. kolovoza 2017.	26-27
9.0	Pie	6. kolovoza 2018.	28
10.0	Q	11. ožujka 2019.	29

Slika 4.2 *Android inačice*

Trenutno je Android jedan od najoptimiziranijih mobilnih operacijskih sustava koji se svakodnevno poboljšava, te svaka sljedeća inačica donosi novitete. Rijetkost je u današnje vrijeme, sresti uređaj s Android inačicom ispod 5.0 Lollipop. Za rad ove aplikacije potreban je API 19, koji podržava već Android 4.4, s pokrivenosti od 99% ukupnog broja mobilnih uređaja, tako da korisnici neće imati problem s pokretanjem aplikacije na svom uređaju.

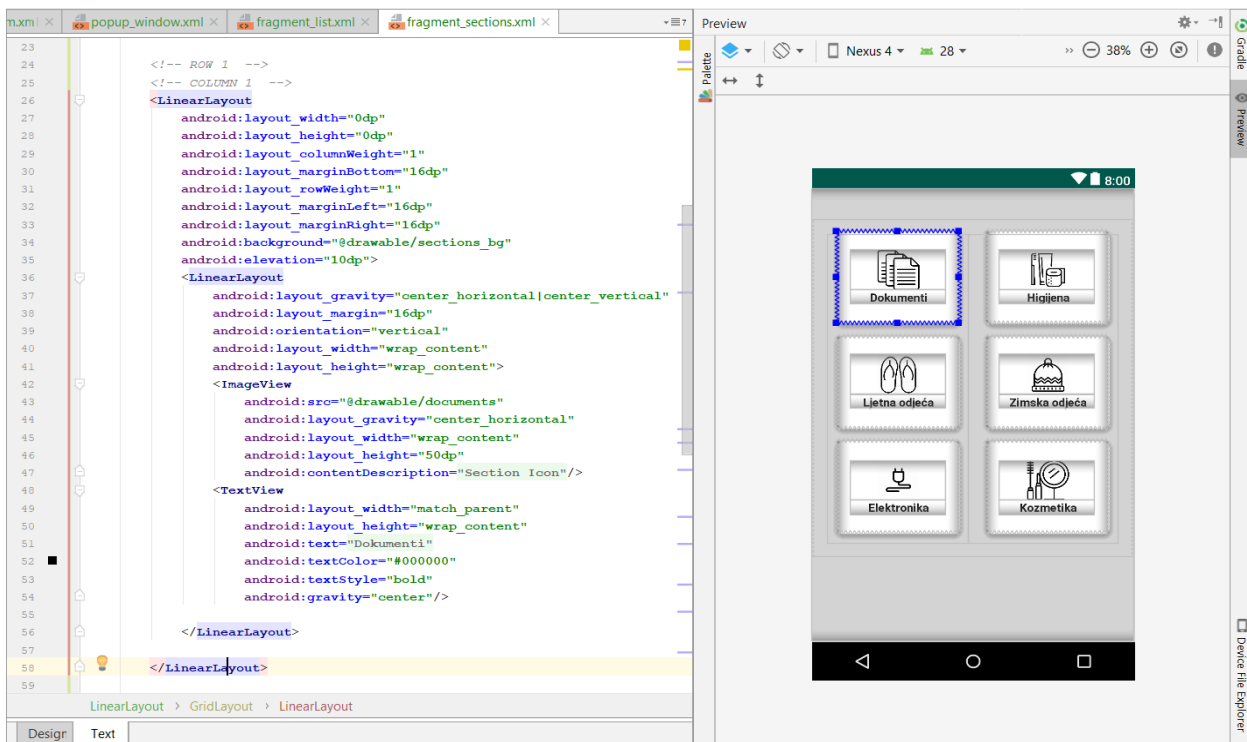
4.2. Razvojne okoline, tehnologije i jezici

Prilikom izrade mobilne aplikacije korištene su sljedeće razvojne okoline, tehnologije i jezici. Za izradu aplikacije koristi se Android Studio, u kojemu se koriste Java i XML. Za obradu slika se koriste Photoshop, dok se za bazu podataka koristi FireBase. Za ostale sitnice su korištene još poneke tehnologije, no najvažnije su navedene u nastavku.

4.2.1. Android Studio

Android Studio okruženje je u kojemu se razvijaju aplikacije za Android platformu. Predstavljen je 16. 05. 2013. na Google- I/O konferenciji. Bazira se na JetBrains IntelliJ IDEA programskoj podršci i ima dizajn specifičan za razvoj Android aplikacija. Dostupne su inačice za Windows, macOS i Linux operativni sustav, dok se u izradu ove aplikacije radila u Android Studio-u na operativnom sustavu Linux, zbog mnogo veće brzine kompiliranja. Mogućnosti koje nudi Android Studio su:

restrukturiranje i brzo uočavanje greški, te ispravljanje koda, Gradle podrška izgradnje, integracije ProGuard-a te mogućnost korisničkog upisa unutar aplikacija, Lint alati za zahvat performansi, iskoristivosti, kompatibilnih inačica i druge probleme, Android virtualne uređaje na kojima je moguće pokretati i ispravljati pogreške u aplikacijama i mnogih drugih. Na slici 4.3 vidljiv je izgled okruženja Android Studia.



Slika 4.3 Okruženje Android Studio

4.2.2. Java

Java je jezik za programiranje razvijen od strane James Gosling, Patrick Naughton i drugih inženjera iz tvrtke Sun Microsystems. Razvoj su započeli 1991. godine, a objavili su ga u 11. mjesecu 1995. godine. Tvrtka Sun posjeduje zaštitni znak na ime Java, ali Java se besplatno može preuzeti s internetskih stranica Sun Microsystems-a. U odnosu na programske jezike do tada, imala je veliku prednost jer se mogla pokrenuti na svakom uređaju za koji postoji JVM (Java Virtual Machine), dok je ostale programe napisane npr. u C-u bilo potrebno prilagoditi platformi na kojoj se izvodi. Kao odgovor na alternative otvorenog koda u literaturi [4] se navodi da je Microsoft morao razviti svoj C# i .NET platformu, jer se java pokazala najboljim izborom za razvoj svih aplikacija. Nalazi se među najkorištenijim programskim jezicima. Izvješća procjenjuju broj korisnika na 7 do više od 10 milijuna.

U današnje vrijeme, Java se upotrebljava na veliko ondje gdje je bitnije brže razvijati programske sustave od zahtjeva za brzinom rada programa. Iako joj je inspiracija C, Java ima veći nivo sigurnosti i pouzdanija je zbog VirtualMachine-a i hermetički zatvorenog okoliša u kojemu svaki program se vrši: Java se koristi za brži razvoj programa s manje grešaka. Upravo to je razlog popularnosti za razvoj programa na mobilnim uređajima te kod financijskih kompanija. Nametnula se kao osnovni jezik Googleovog sustava Android, iako joj u posljednje vrijeme konkurira Kotlin.

4.2.3. Opisni jezik XML

XML je skraćenica za EXtensible Markup Language – tj. jezik za označavanje podataka. Cilj je bio stvaranje jednog jezika koji je jednostavan, razumljiv i ljudskim bićima i računalnom programu. Postupak realizacije je veoma jasan: sadržaj se uokviruje određenim oznakama koje ga opisuju i imaju poznato, ili prosto shvatljivo značenje. Oznake i njihov format dosta nalikuju HTML-u. Danas je XML vrlo rasprostranjen i koristi se za različite primjene kao što su: razdvajanje podataka i prezentacija, razmjena podataka, pohrana podataka, činenje podataka dostupnijim, te izrada novih jezika koji su specijalizirani za označavanje. Standardiziran je, a za njegovo standardiziranje zaduženo je udruženje ustanova World Wide Weba. U programskom kodu 4.1 vidljiv je prikaz osnovnih naredbi koje se koriste za raspored elemenata unutar Android Studia.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

    <TextView
        android:layout_width="fill_parent"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

    <Button
        android:id="@+id/button1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="@string/hello" />

</LinearLayout>
```

Programski kod 4.1 *Primjer XML naredbi u Android Studio-u*

4.2.4. Grafički program Photoshop

Adobe Photoshop, ili skraćeno Photoshop, je grafički program za računala, kojega je razvila i izdala tvrtka iz Amerike pod nazivom Adobe System. Trenutno je svjetski najpopularniji program za uređivanje slika. Prva inačica programa Photoshop 1.0 napravljena je 1990. godine, dok je posljednja inačica, Photoshop CS6 trinaesta generacija istog proizvoda. Uključuje izvanredan napredak u performansama koje pomiču granice profesionalne digitalne obrade slike, te donosi nove i unaprijeđene alate za dizajn. Adobe Mercury Graphics Engine koristi podršku različitih 64-bitnih platformi za ubrzavanje obrada i uređivanje slika, te brže procesiranje većih slika, dok je zahvaljujući novim i unaprijeđenim video alatima omogućeno i stvaranje videozapisa. Tijekom izrade ove aplikacije, Photoshop je korišten za obradu slika, izradu dugmadi, loga aplikacije, kao i uređivanje naslovne stranice. Na slici 4.4 je zaslon koji se koristi kao splash zaslon, a izrađen je unutar programa Photoshop.



Slika 4.4 Splash zaslon aplikacije

4.2.5. Firebase

Firebase je prvotno napravila tvrtka iz San Francisca koja pruža pozadinske usluge, te računarstvo u oblaku. Veliki broj aplikacija za mobilne i web aplikacije koristi usluge ove tvrtke. Andrew Lee i James Tamplin 2011. godine su osnovali FireBase, te su počeli raditi u travnju 2012. godine. Osnovni proizvod im je baza podataka za rad u stvarnom vremenu, sučelje za programiranje aplikacija (API) koje programerima omogućuje pohranjivanje i sinkronizaciju podataka između različitih klijenata. Tvrtku je 2014. godine kupio Google, te je danas jedan od najčešćih izbora prilikom rada s bazom podataka, tako da će i u ovoj aplikaciji biti korištene usluge Firebase-a. Prednosti korištenja Firebase-a su mnogobrojne, a među ostalima izdvajaju se te da nije potrebno brinuti o upravljanju serverom, ima podršku za web, iOS, te naravno Android uređaje. U aplikaciju se uključuje preko biblioteka, a budući da ima brojne značajne usluge, u nastavku će biti navedeno par: Firebase Analytics, Realtime Database, Storage, Hosting, Crash Reporting, Firebase Authentication. Pomoću navedenih usluga otvaraju se brojne mogućnosti, poput pregleda broja aktivnih korisnika u posljednjih N dana, broju koji nam govori koliko se puta aplikacija srušila, te ukoliko postoje prihodi od aplikacije vidimo i njih. Za našu aplikaciju potrebni su Firebase Authentication i Realtime DataBase.

4.3. Programske komponente i razvoj mobilne aplikacije

Budući da je aplikacija složena iz više dijelova, te zbog njene kompleksnosti, u nastavku su objašnjeni neki najvažniji dijelovi s kojima se postigla ušteda resursa, odnosno rad aplikacije. Proces izrade prati izradu aktivnosti, njihovih funkcionalnosti, unos u Manifestu, pokazuje način spremanja, te rada s online bazom podataka, korištenje reciklirajući elemenata, kao i upotrebu ostalih mogućnosti.

4.3.1. Android Manifest

Android Manifest je datoteka pisana XML opisnim jezikom, te je datoteka koju svaka aplikacija mora imati. Opisuje osnovne informacije o aplikaciji operacijskom sustavu, Google Play trgovini, kao i Android alatima za izradu. Unutar nje su opisane sve aktivnosti, servisi i davatelji sadržaja. Pokazuje kako koja komponenta može biti pokrenuta, te deklarira mogućnosti pokretanja aplikacije ovisno o uređaju. Unutar ove datoteke navedeno je koje dozvole sustava aplikacija koristi, a za izradu ove aplikacije potrebna je samo dozvola za internet, koja je potrebna da bi baza podataka mogla raditi.

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.menzal">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="Menzal"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportsRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".IzborJela"
13            android:noHistory="true"
14            ></activity>
15        <activity
16            android:name=".ProfilActivity"
17            android:noHistory="true" />
18        <activity
19            android:name=".PodatciActivity"
20            android:theme="@style/Theme.AppCompat.Light.NoActionBar" />
21        <activity android:name=".MainActivity">
22            <intent-filter>
23                <action android:name="android.intent.action.MAIN" />
24
25                <category android:name="android.intent.category.LAUNCHER" />
26            </intent-filter>
27        </activity>
28    </application>
29
30 </manifest>
```

Slika 4.5 Izgled datoteke AndroidManifest.xml

4.3.2. Aktivnosti

Aktivnost je osnovna komponenta svih Android aplikacija, te ih svaka aplikacija mora sadržavati. Osnovne aplikacije sadrže samo jednu aktivnost, ali većina aplikacija jednostavno ima potrebu za većim brojem aktivnosti. Jedan zaslon aplikacije bi se mogao predstaviti s jednom aktivnosti, međutim prilikom izrade ove aplikacije da se koriste samo aktivnosti moralo bi ih biti previše, pa se koriste neki drugi elementi poput fragmenata. Aplikacija se sastoji od 4 aktivnosti što se i vidi na slici 4.5 u Android Manifestu. MainActivity je glavna aktivnost, te se ona prva pokreće, međutim ona služi samo za provjeru i spremanje podataka. Budući da ova metoda nema neke funkcionalnosti koje korisnik može vidjeti, njemu se prikaže takozvani Splash zaslon, odnosno kratka slika s logom aplikacije, koja traje jednu sekundu, dok se u pozadini odvija provjera, odnosno spremanje podataka. Metoda koja se prva poziva je metoda onCreate i u njoj je postavljen vremenski sklop na jednu sekundu.


```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    inicijalizacija();
    Toast.makeText(context: this, text: "prvi", Toast.LENGTH_SHORT).show();

    final Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            pocetniw.performClick();
        }
    }, delayMillis: 1000);
}

@Override
protected void onResume() {
    super.onResume();
    final Handler handler = new Handler();
    handler.postDelayed(new Runnable() {
        @Override
        public void run() {
            pocetniw.performClick();
        }
    }, delayMillis: 1000);
}

```

Slika 4.6 Metode prve aktivnosti - MainActivity-a

Nakon sekunde događa se klik na sliku koji uzrokuje provjeru da li postoje spremljeni podatci, te budući da prvi put ne postoje, poziva novu aktivnost Podatci.

```

pocetniaw.setOnClickListener((v) -> {
    SharedPreferences userInfosharedPrefs = getSharedPreferences( name: "userInfo", MODE_PRIVATE);
    SharedPreferences.Editor ed=userInfosharedPrefs.edit();
    SharedPreferences kalorijesharedPref= getSharedPreferences( name: "kalorijeInfo", Context.MODE_PRIVATE);
    SharedPreferences.Editor KCalEditor= kalorijesharedPref.edit();
    SharedPreferences prijenosSharedPref;

    String datumSad = new SimpleDateFormat( pattern: "dd-MM-yyyy", Locale.getDefault()).format(new Date());
    String datumStari;

    if(userInfosharedPrefs.contains("initialized")){...}
    else {
        Intent intent=new Intent( packageContext: MainActivity.this,PodatciActivity.class);
        MainActivity.this.startActivity(intent);
    }
});

```

Slika 4.7 Metoda nakon Splash zaslona-a

Nakon pokretanja aktivnost Podatci, prikazuje se zaslون kao na slici 4.8, gdje korisnik mora popuniti sve podatke, što je uvjetovano metodom prikazanom na slici 4.9. Dok svi podatci ne budu uneseni korisniku se neće omogućiti nastavak. Ti uvjeti se provjeravaju na način, da se provjeri svaka ćelija u kojoj je moguć unos, da li je prazna. Ranije su već ograničene na brođčane vrijednosti, one ćelije koje smiju imati samo brođčanu vrijednost.

Slika 4.8 Izgled aktivnosti PodatciActivity

```

if (et_godine.getText().toString().isEmpty() || et_ime.getText().toString().isEmpty() ||
    et_masa.getText().toString().isEmpty() ||
    et_visina.getText().toString().isEmpty() ||
    rg_aktivnost.getCheckedRadioButtonId()==-1 ||
    rg_spol.getCheckedRadioButtonId()==-1 ||
    rg_zelja.getCheckedRadioButtonId()==-1){
    Toast.makeText(context: PodatciActivity.this, text: "Popunite sve podatke", Toast.LENGTH_SHORT).show();
}

```

Slika 4.9 Metoda koja provjera da li su svi podatci uneseni

Nakon što su svi podatci uneseni, aplikacija ih obrađuje, te u ovisnosti o parametrima poput spola, aktivnosti i želje, računa broj kalorija koji je dozvoljen za dnevni unos, što se vidi na slici 4.10. Unutar if uvjeta je postavljeno da se provjerava da li radio gumb koji je odabran predstavlja muško ili žensko. Ukoliko se radi o muškoj osobi, poziva se izračun kalorija koji koristi izraze (2-1) i (2-3).

```

float visina, masa;
int godine;

visina=Float.parseFloat(et_visina.getText().toString());
masa=Float.parseFloat(et_masa.getText().toString());
godine=Integer.parseInt(et_godine.getText().toString());

if (rg_aktivnost.getCheckedRadioButtonId()==rb_akt1.getId()){aktivnost=1.2f;}
else if (rg_aktivnost.getCheckedRadioButtonId()==rb_akt2.getId()){aktivnost=1.375f;}
else if (rg_aktivnost.getCheckedRadioButtonId()==rb_akt3.getId()){aktivnost=1.55f;}
else if (rg_aktivnost.getCheckedRadioButtonId()==rb_akt4.getId()){aktivnost=1.725f;}
else if (rg_aktivnost.getCheckedRadioButtonId()==rb_akt5.getId()){aktivnost=1.9f;}

if(rg_zelja.getCheckedRadioButtonId()==rb_smrsati.getId()){zelja=-500;}
else if(rg_zelja.getCheckedRadioButtonId()==rb_odrzavati.getId()){zelja=0;}
else{zelja=500;}

if (rg_spol.getCheckedRadioButtonId()== rb_musko.getId()){
    spol="Muško";

    idealna_masa=50+2.3f*(visina-152.4f)/2.54f;
    brojkalorija=88.362f+13.397f*masa+4.799f*visina-5.677f*godine;
    brojkalorija=brojkalorija*aktivnost+zelja;
}
else{
    spol="Žensko";
    idealna_masa=45.5f+2.3f*(visina-152.4f)/2.54f;
    brojkalorija=447.593f+9.247f*masa+3.098f*visina-4.330f*godine;
    brojkalorija=brojkalorija*aktivnost+zelja;
}

```

Slika 4.10 Izračun dnevnog unosa kalorija

4.3.3. SharedPreferences

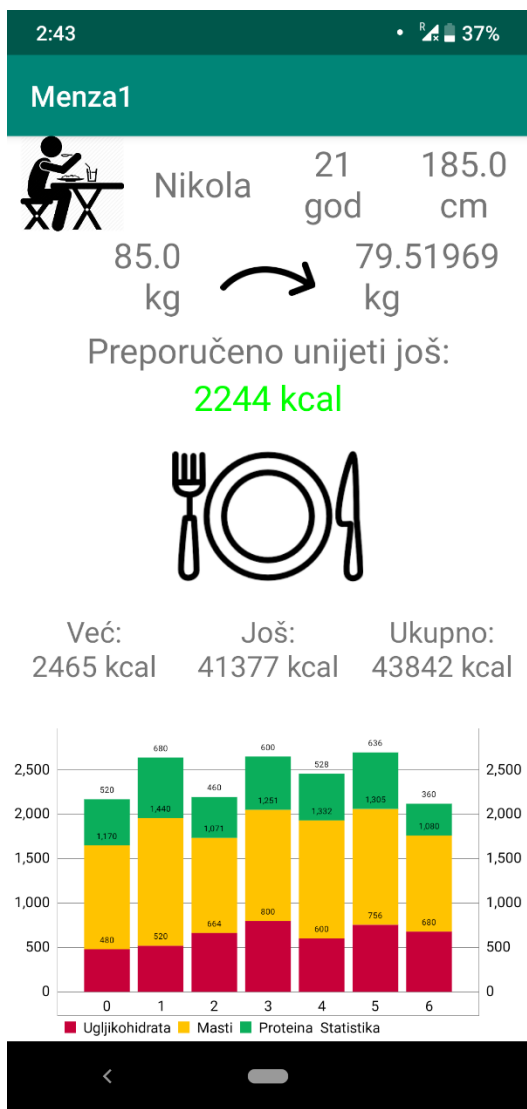
SharedPreferences ili dijeljene postavke pohranjuju parove u obliku ključ i vrijednost. Ključ je tipa String, a vrijednost bilo kojeg od osnovnih oblika tipa podatka. Mode ograničava pravo pristupa na privatno ili javno. Podaci koji se pohranjuju kao SharedPreferences održavaju se dok se aplikacija izvodi, te prilikom ostalih pokretanja aplikacije. Budući da korisnik neće uvijek imati pristup internetu, SharedPreferences se koristi za spremanje njegovih podataka, i svih podataka neophodnih za rad aplikacije bez interneta.

```
private void spremi_podatke() {  
    String datumpocetka = new SimpleDateFormat( pattern: "dd-MM-yyyy", Locale.getDefault()).format(new Date());  
  
    SharedPreferences userInfoSharedPref= getSharedPreferences( name: "userInfo", Context.MODE_PRIVATE);  
    SharedPreferences kalorijskiSharedPref= getSharedPreferences( name: "kalorijeInfo", Context.MODE_PRIVATE);  
    SharedPreferences.Editor uEditor= userInfoSharedPref.edit();  
    SharedPreferences.Editor kCalEditor= kalorijskiSharedPref.edit();  
  
    uEditor.putString("ime", et_ime.getText().toString());  
    uEditor.putBoolean("initializirano", true);  
    uEditor.putFloat("godine", Float.parseFloat(et_godine.getText().toString()));  
    uEditor.putFloat("visina", Float.parseFloat(et_visina.getText().toString()));  
    uEditor.putFloat("masa", Float.parseFloat(et_masa.getText().toString()));  
    uEditor.putString("spol", spol);  
    uEditor.putFloat("aktivnost", aktivnost);  
    uEditor.putFloat("idealna_masa", idealna_masa);  
    uEditor.putFloat("broj_kalorija", brojkalorija);  
    uEditor.putFloat("standardkcal", brojkalorija);  
    uEditor.putFloat("ukupnozapromijeniti", razlikamasa*8000);  
    uEditor.putFloat("joszapromijeniti", razlikamasa*8000);  
    uEditor.putFloat("dosadpromijenjeno", 0);  
    uEditor.putInt("zelja", zelja);  
    uEditor.putString("datumpocetka", datumpocetka);  
    uEditor.putString("datumzadnji", datumpocetka);  
  
    uEditor.apply();  
  
    kCalEditor.putInt("brojac", 1);  
    kCalEditor.putString("pocetak", datumpocetka);  
    kCalEditor.putFloat("dozvoljenoKalorija", brojkalorija);  
    kCalEditor.putString("datum "+String.valueOf(1), datumpocetka);  
    kCalEditor.putInt("kalorija "+String.valueOf(1), 0);  
  
    kCalEditor.apply();  
}
```

Slika 4.11 Prikaz upotrebe SharedPreferences

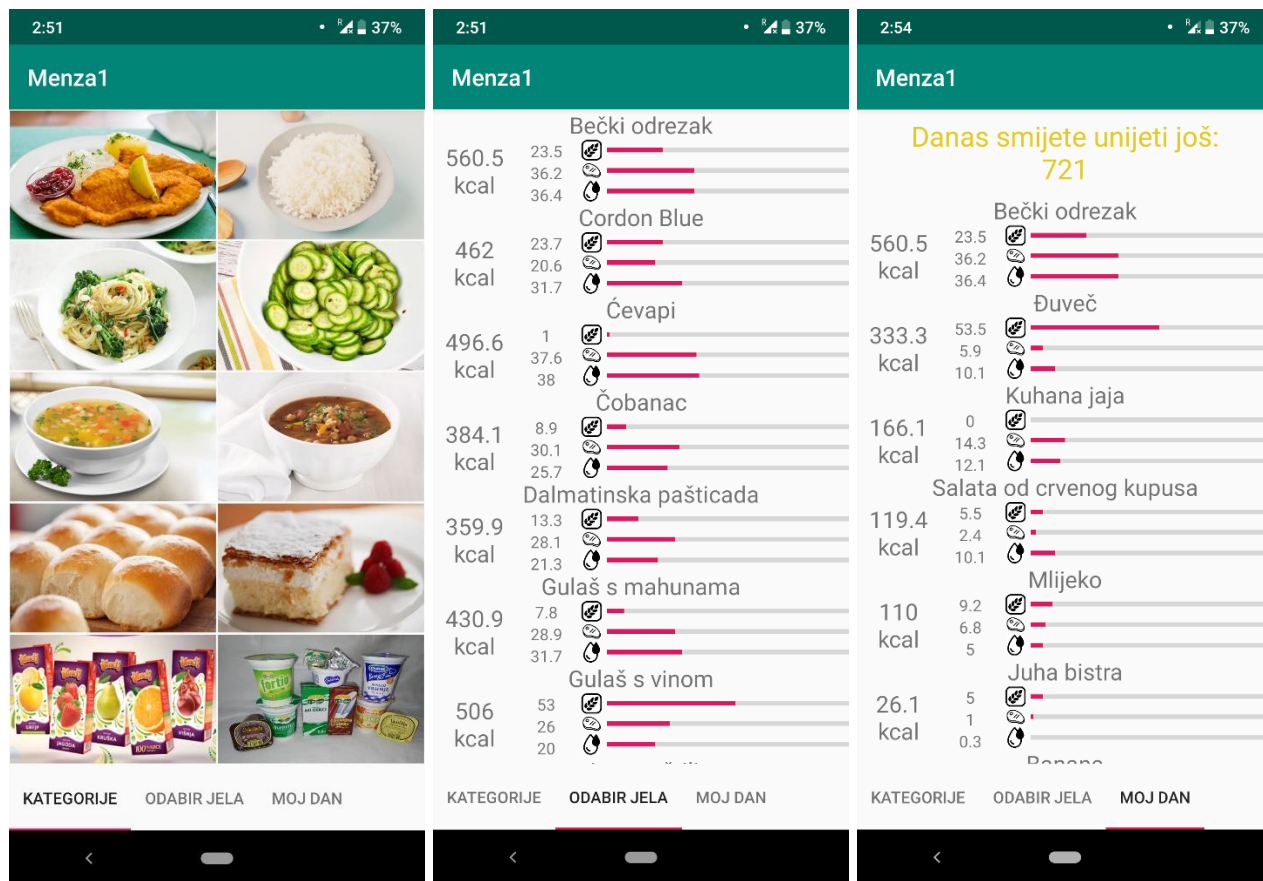
Pomoću SharedPreferences postignuto je očuvanje dodanih podataka u aplikaciji. Metoda spremi_podatke sa slike 4.11 prikazuje način spremanja koristeći SharedPreferences. U ovom dijelu programa naprave se dvije datoteke u lokalnoj memoriji pod nazivi userInfo i kalorijeInfo. Prva sadrži osnovne informacije o korisniku, dok druga sadrži osnovne informacije o kalorijama.

Nakon što se spremi, pokreće se aktivnost Profil, koja ima izgled kao na slici 4.12. Aktivnost ProfilActivity je aktivnost koja se pokreće prilikom svakog sljedećega pokretanja aplikacije, te prikazuje spremljene podatke. U gornjem dijelu su prikazane osnovne informacije o korisniku, dok sredina prikazuje preostali preporučeni dnevni unos, a na dnu je vidljiv napredak kroz zadnje dane. Graf je stupčani, te koristi podatke iz online baze.



Slika 4.12 Izgled aktivnosti ProfilActivity

Iz te aktivnosti je moguće prijeći na aktivnost zvanu IzborJelaActivity, koja je zapravo i najsloženija i ima sljedeći izgled (slika 4.13). Na prvom zaslonu potrebno je izabrati kategoriju u ovisnosti od koje se formira ponuda na drugom zaslonu, a odabir stavki na drugom utječe na treći. Kraj svake stavke je grafički prikazan omjer ugljikohidrata, masti i proteina, radi lakše predodžbe korisniku. Zaslon ove aktivnosti se sastoji od TabLayouta, ViewPagera koji sadrže u sebi fragmente koji opet svaki u sebi sadrže RecyclerView. Na osnovu ovoga je ušteda od par aktivnosti, svakako ubrzala rad aplikacije.



Slika 4.13 Izgled IzborJela aktivnosti

Na prvom zaslonu je potrebno odabrati jednu od deset ponuđenih kategorija. U ovisnosti o izboru se postavlja zaslon broj dva. Na slici 4.14 se vidi način određivanja kategorije pomoću SharedPreferences.

```

iw_glavnajela.setOnClickListener((v) -> {
    editorKat.putInt("izborkategorije",0);
    editorKat.apply();
    tabs.getTabAt(index: 1).select();
});

iw_prilozi.setOnClickListener((v) -> {
    editorKat.putInt("izborkategorije",1);
    editorKat.apply();
    tabs.getTabAt(index: 1).select();
});

iw_vegejela.setOnClickListener((v) -> {
    tabs.getTabAt(index: 1).select();
    editorKat.putInt("izborkategorije",2);
    editorKat.apply();
});

```

Slika 4.14 Odabir kategorije

Da bi se na drugom zaslonu znalo koja će jela biti prikazana, potrebno je iz spremljene datoteke SharedPreferences, povući koja je kategorija odabrana, taj broj se sprema u novu cjelobrojnu varijablu izborkat, te se zatim u ovisnosti o njoj postavljaju nazivi jela, te njihove kalorijske i nutritivne vrijednosti. Slike 4.14 i 4.15 su prikazale postupke samo za prva 3 slučaja, ali postupak je isti i za ostale. Nazivi, kalorijske i nutritivne vrijednosti svih jela su spremljene u datoteci strings.xml koja se koristi za spremanje svih stringova na jednom mjestu. Kada se dohvaćaju brožčane vrijednosti potrebno je izvršiti parsiranje odnosno pretvorbu iz tipa String u tip int ili float ovisno od situacije.

```

int izborkat= odabirKat.getInt(key: "izborkategorije", defValue: 0);

if(izborkat==0){//glavna jela
    nazivi=getResources().getStringArray(R.array.glavnajelaNazivi);
    ugljici=getResources().getStringArray(R.array.glavnajelaUgljiko);
    masti=getResources().getStringArray(R.array.glavnajelaMasti);
    proteini=getResources().getStringArray(R.array.glavnajelaProteini);
    kalorije=getResources().getStringArray(R.array.glavnajelaKalorije);
}
else if(izborkat==1){
    //prilozi
    nazivi=getResources().getStringArray(R.array.priloziNazivi);
    ugljici=getResources().getStringArray(R.array.priloziUgljici);
    masti=getResources().getStringArray(R.array.priloziMasti);
    proteini=getResources().getStringArray(R.array.priloziProteini);
    kalorije=getResources().getStringArray(R.array.priloziKalorije);
}
else if(izborkat==2){//vegejela
    nazivi=getResources().getStringArray(R.array.vegejelaNazivi);
    ugljici=getResources().getStringArray(R.array.vegejelaUgljici);
    masti=getResources().getStringArray(R.array.vegejelaMasti);
    proteini=getResources().getStringArray(R.array.vegejelaProteini);
    kalorije=getResources().getStringArray(R.array.vegejelaKalorije);
}

```

Slika 4.15 Postavljanje određene podkategorije

4.3.4. RecyclerView

Ova komponenta koristi se kada je potrebno prikazati veliku količinu podataka u obliku liste koju mora biti moguće pomjerati gore ili dolje. RecyclerView odlikuje recikliranje svojih elemenata zbog čega se povećava brzina rada samih komponenti. Literatura[2] navodi da sličnu funkcionalnost posjeduje i ListView, ali zbog potrebe da se recikliranje implementira od strane korisnika polako se izbacuje iz uporabe. Dodatna prednost RecyclerViewa nad ListViewom je korištenje LayoutManagera preko kojega se može direktno utjecati na raspored i ponašanje elemenata liste. Recikliranje elemenata znači da se na zaslonu prikazuje samo onoliko elemenata koliko može stati, uz dodatnih par elemenata prije i poslije, radi ugladenog klizanja.

```
recyclerView2=v.findViewById(R.id.recyclercic2);
recyclerView2.setLayoutManager(new LinearLayoutManager(getActivity()));

adapter2=new RecyclerViewAdapter2( listener: this);

adapter2.addData(nazivi,ugljici,proteini,masti,kalorije);
recyclerView2.setAdapter(adapter2);
```

Slika 4.16 Prikaz RecyclerViewa

RecyclerView se koristi na drugom i trećem zaslonu aktivnosti IzborJela, te se na slici 4.16 vidi njegovo postavljanje. Da bi se postavio potrebno je pronaći njegov View preko id-a, postaviti upravitelja rasporeda (linearni ili relativni), dodati podatke u novokreirani adapter, te adapter postaviti kao adapter RecyclerViewa. RecyclerView zahtijeva jasno definirane podatke, a on dalje brine o njihovom recikliranju.

4.3.5. Fragmenti

Fragmenti, prema [1] predstavljaju klasu koja omogućuje modularan dizajn Activitya. Riječ je o dodatnom sloju smještenom između sučelja i Activitya. Moguće ih je koristiti više unutar jednog Activitya, ali i koristiti ih kroz više Activitya. Imaju vlastiti životni ciklus koji je usko vezan uz Activity u kojem se nalaze. Tijekom životnog ciklusa Activitya svi fragmenti prolaze iste tranzicije, a dok je Activity aktivan moguće je manipulirati pojedinim fragmentima. Rad s fragmentima nije jednostavan, a prilikom njihova korištenja potrebno je biti oprezan. Za upravljanje fragmentima koristi

se klasa PagerAdapter u kojoj se nalazi metoda getItem koja služi za navigaciju između tabova. Fragmenti se koriste da bi se s jednim Activity-em moglo upravljati više prikaza. Activity koriste mnogo više resursa nego fragmenti, pa su fragmenti mnogo isplativiji.

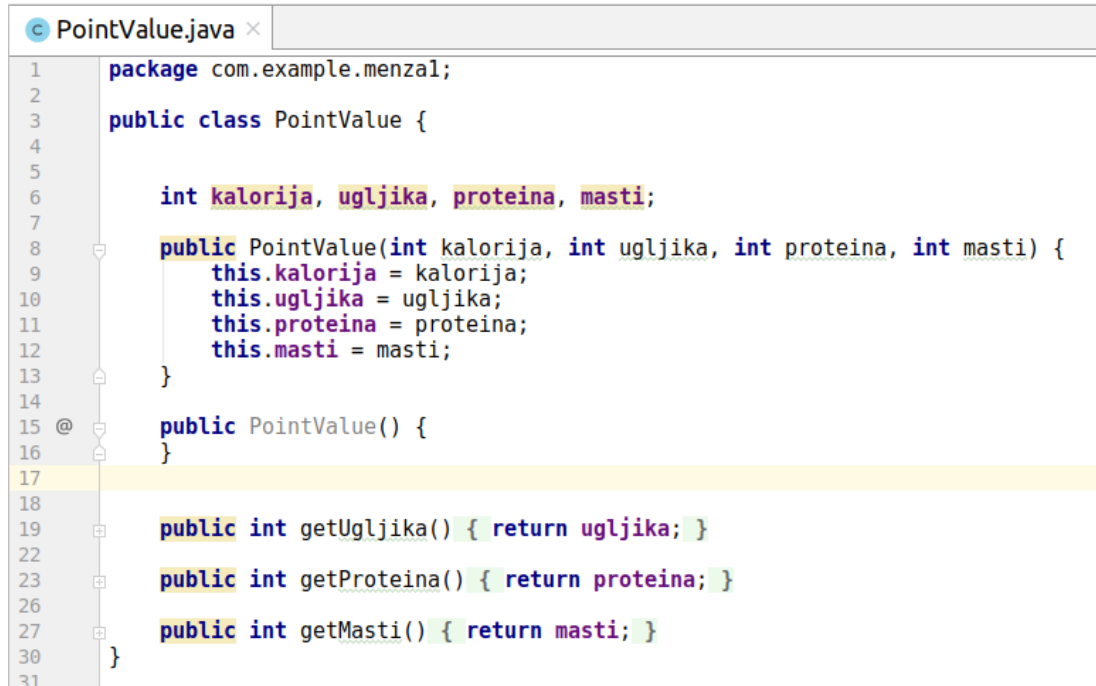
4.3.6. FireBase

Budući da je aplikacija predviđena za rad bez interneta, baza podataka se koristi u situacijama kada postoji internet. Da bi se FireBase koristio potrebno ga je implementirati što je u novijim inačicama Android Studia automatizirano. Na slici je prikazan objekat baze podataka, te se vidi da se poziva datoteka s imenom podatci.

```
database=FirebaseDatabase.getInstance();  
reference=database.getReference( path: "podatci");
```

Slika 4.17 Kreiranje objekta Firebase baze podataka

Klasa PointValue je jednostavna klasa koja sadrži varijable za nutritivne vrijednosti, te se koristi za stvaranje objekata koji će se spremati u bazu podataka.



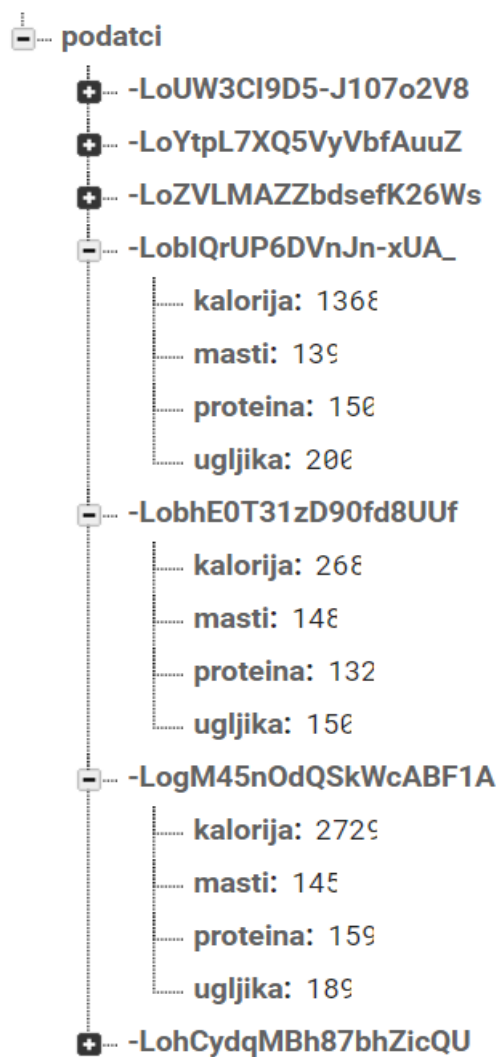
```
1 package com.example.menzal;  
2  
3 public class PointValue {  
4  
5  
6     int kalorija, ugljika, proteina, masti;  
7  
8     public PointValue(int kalorija, int ugljika, int proteina, int masti) {  
9         this.kalorija = kalorija;  
10        this.ugljika = ugljika;  
11        this.proteina = proteina;  
12        this.masti = masti;  
13    }  
14  
15    @  
16    public PointValue() {  
17    }  
18  
19    public int getUgljika() { return ugljika; }  
22  
23    public int getProteina() { return proteina; }  
26  
27    public int getMasti() { return masti; }  
30 }  
31
```

Slika 4.18 Klasa PointValue

Nakon kreiranja objekta klase `PointValue`, ubacuje ga se u bazu podataka (slika 4.19), dok izgled u bazi vidimo na slici 4.20.

```
String id= reference.push().getKey();  
PointValue podatak=new PointValue(kal,uglj,prot,mast);  
reference.child(id).setValue(podatak);
```

Slika 4.19 Ubacivanje podataka u online bazu



4.20 Model baze podataka

5. PRIKAZ KORIŠTENJA I ISPITIVANJE APLIKACIJE SA ANALIZOM REZULTATA

U ovom dijelu rada bit će ispitane funkcionalnosti aplikacije, te analiza rezultata. Budući da se u aplikaciji mora napraviti izbor kod želje, bit će izvršena 3 ispitivanja. U prvom slučaju osoba želi smršaviti, u drugom udebljati se i u trećem želi ostati na istoj kilaži.

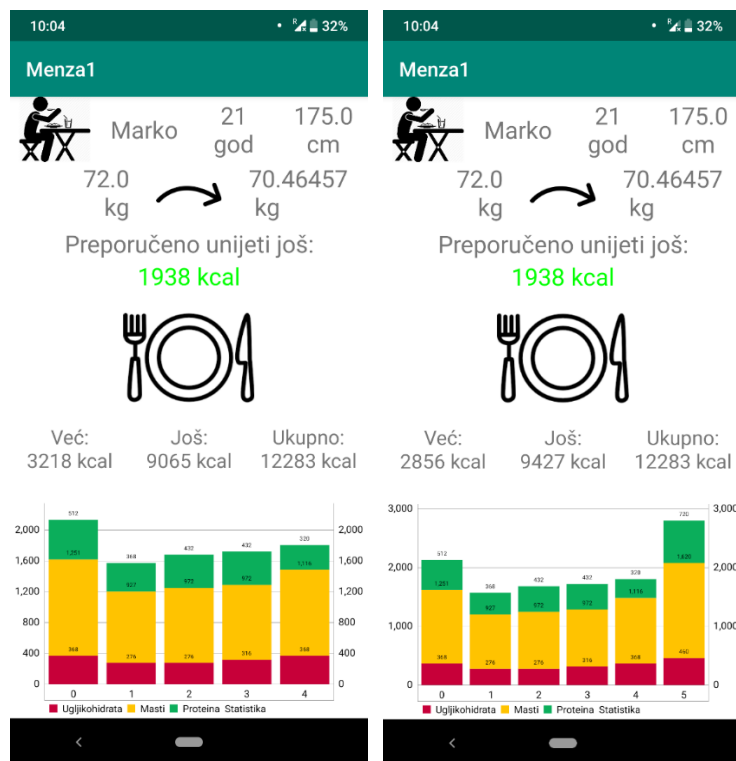
5.1. Ispitivanje funkcionalnosti aplikacije

Unutar ovoga dijela ispitana su tri slučaja studenata s potrebom za ovom aplikacijom. Unutar aplikacije se unose njihovi podaci i njihove želje. Aplikacija preporučuje njihove potrebe, međutim prilagođava se njihovim željama. Ovaj dio je koristan osobama koje izgledaju neproporcionalno, pa iako imaju preveliku masu nisu debeli (npr. zbog izuzetne visine). Također je bitno pretpostaviti, da korisnik ako unese dnevno ograničenje kalorija.

5.1.1. Prvo ispitivanje, slučaj kalorijskog deficita

U prvom ispitivanju uneseni su podaci studenta Marko, koji je visok 175 centimetara, ima 21 godinu, malu aktivnost i 72 kilograma. Marko želi izgubiti na tjelesnoj masi, te je odabrao tu mogućnost. Ovo ispitivanje bi trebalo rezultirati kalorijskim deficitom, te preporučenim dnevnim unosom od 1938 kcal.

Prvo ispitivanje je očekivano prepoznalo potrebu za kalorijskim deficitom, te kao takvo postavilo dnevni unos kalorija na manju vrijednost od preporučene za održavanje tjelesne mase. Preporučeni unos kalorija na dnevnoj bazi za Marka je 1938 kcal, a ukupan broj kalorija koje treba izgubiti je 12 283 kcal. Na slici 5.1 se vidi da nakon 5 dana korištenja aplikacije Marko je izgubio 3218 kcal, a preostalo mu je još 9 065 kcal. Sljedeći dan je prekoračio dnevno ograničenje pa su ove brojke spustile se odnosno povećale na 2856 kcal i 9427 kcal. U ovom ispitivanju potvrđeno je da aplikacija radi ispravno i točno.

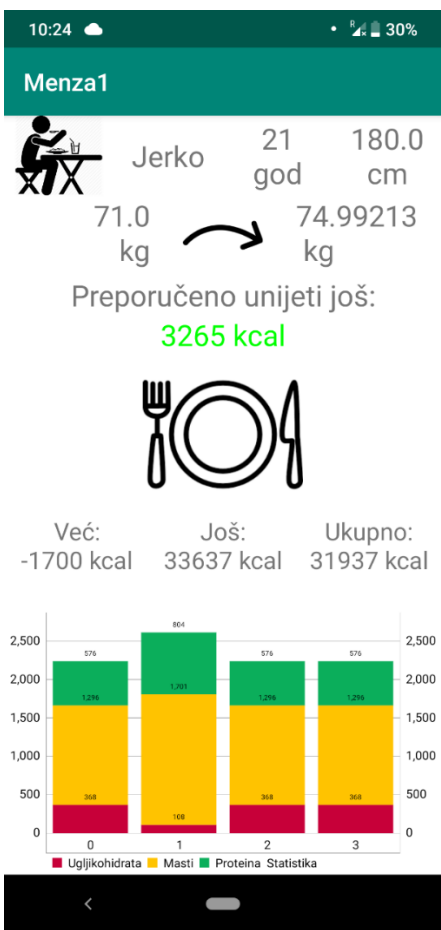


Slika 5.1 Student Marko, prvi slučaj

5.1.2. Drugo ispitivanje, slučaj kalorijskog suficita

U drugom ispitivanju student Jerko ima 180 centimetara visine, 22 godine i 71 kilogram masu. Aktivnost mu je umjerenjena i želi se udebljati. Kod ovoga ispitivanja bi rezultat trebao biti kalorijski suficit s preporučenim dnevnim unosom od 3265 kcal.

Jerko je izrazio želju za dobivanjem tjelesne mase, pa je na osnovu njegovih podataka izračunato da bi morao unositi svakodnevno 3265 kcal, da bi bio u kalorijskom suficitu. Ukupan broj kilograma koje bi trebao dobiti izražen u kalorijama iznosi 31 397 kcal, međutim vidljivo na slici 5.2 je da on svakodnevno unosi manje od preporučenog unosa, te da si tim načinom produljuje vrijeme potrebno za udebljati se. Nakon tjedan dana ukoliko se bude nalazio u lošijoj situaciji, nego kada je krenuo, aplikacija će mu preporučiti da pokuša ponovno i dobro preispita svoje podatke, jer velika greška može nastupiti ukoliko se precijeni aktivnost, što je ovaj student vjerojatno i napravio. Ovo ispitivanje prikazuje ponašanje aplikacije, ukoliko joj se unose drugačiji podatci od onih koje ona očekuje, te njena reakcija na pogrešne podatke, odnosno ne pridržavanje njenih uputa.



Slika 5.2 Drugi slučaj, student Jerko

5.1.3. Treći slučaj, kalorijska nula

Treći slučaj je studentica Ivana koja ima 59 kilograma, visoka je 164 centimetara, ima 22 godine i veliku aktivnost i želi održavati trenutnu masu. Kod ovoga ispitivanja je bitno da nema ni kalorijskog suficita, ni deficita, pa je tako njen kalorijski dnevni unos jednak 2425 kcal. Treće ispitivanje je prepoznalo da trenutna masa i idealna masa kod ove studentice nisu isti, ali se prilagodio njenoj želji. Iako bi prema proračunima, korisnica trebala izgubiti koji kilogram, budući da je njena želja ostati na istoj tjelesnoj masi, aplikacija neće preporučiti ni kalorijski suficit ni deficit. Omogućit će joj praćenje dnevnog unosa kalorija, te je njen ukupni broj kalorija koje treba promijeniti 0.

5.2. Usporedba rezultata

Ulazni i izlazni podatci prethodna 3 ispitivanja prikazani su tablično u nastavku. Tablica 5.1 prikazuje ulazne podatke, dok se na tablici 5.2 vide izlazni podatci.

Tablica 5.1 *Ulazni podatci*

Student	Godine	Visina [cm]	Masa [kg]	Aktivnost	Želja
Marko	21	175	72	mala	Smršaviti
Jerko	21	180	71	umjerena	Udebljati se
Ivana	22	164	59	velika	održavati

Tablica 5.2 *Izlazni podatci*

Student	Idealna masa [kg]	Potreba	Preporučeni unos [kcal]
Marko	70.46	Smršaviti	1938
Jerko	74.99	Udebljati se	3265
Ivana	56	Smršaviti	2425

Potrebno je usporediti potrebe i želje korisnika, te je to prikazano tablicom 5.3 u nastavku.

Tablica 5.3 *Usporedba želje i potrebe studenata*

Student	Potreba	Želja	Akcija
Marko	Smršaviti	Smršaviti	Kal. Deficit
Jerko	Udebljati se	Udebljati se	Kal. Suficit
Ivana	Smršaviti	Održavati	ništa

Iz tablice 5.3 je vidljivo da će se u prva dva slučaja aplikacija nastaviti izvršavati prema planu, dok u trećem slučaju, želja korisnika se stavlja ispred pretpostavke aplikacije, te se Ivani neće preporučiti kalorijski deficit, nego je promjena u njenim kalorijama jednaka nuli.

6. ZAKLJUČAK

Cilj završnoga rada je bio napraviti aplikaciju, koja će prvotno biti namijenjena studentima, ali i općenito ljudima s područja Hrvatske. U odnosu na ostala rješenja dostupna na internetu, cilj je bio imati ponudu sa što više jela iz studentske menze jer je to najčešći izbor studenata prilikom ishrane. Osim ponude, želja je bila korisniku omogućiti uvid u kalorijske i nutritivne vrijednosti svih obroka koje jede, te da se na taj način pokuša podignuti svijest o zdravom životu.

Prilikom izrade aplikacije korišteno je programsko okruženje Android Studio, jer prema istraživanjima, kako navodi literatura, najveći broj studenata ima Android uređaj. Unutar same aplikacije korištene su razne tehnike kako bi se postiglo što brže izvođenje, bez ogromne veličine aplikacije. Korisnik pri prvom korištenju aplikacije unosi svoje podatke, na osnovu kojih mu se vrši izračun, koji se dalje koristi prilikom svakodnevnih obroka. Za razliku od konkurencije slične aplikacije, za ovaj primjer NutriMenze, omogućeno je spremanje, te pregled proteklih dana i unesenih kalorija. Također, korišteni su reciklirajući elementi, odnosno elementi koji dinamički rade s memorijom, te zauzimaju samo koliko je potrebno. Aplikacija može raditi bez interneta, te joj je to prednost, budući da se lako zaboravi unijeti informacije o jelu, nekoliko sati nakon njega. Međutim, kada ima internet, podatci prethodnih dana se podižu na online bazu podataka.

Ostvareno rješenje aplikacije se pokazalo korisnim, te učinkovitim jer za očekivane ulazne vrijednosti, daje očekivane izlazne rezultate. Aplikacija se pridržava teorijskih podloga, te kao takva nije odgovorna za odstupanja kod ljudskih organizama, koji odstupaju od teorijskih proračuna. Daljnjim razvojem bi se aplikacija mogla unaprijediti, dodavanjem mogućnosti za skeniranje crtičnih kodova, jer je u današnje vrijeme korisnicima bitna brzina unosa podataka. Također detaljnijom teoretskom podlogom bi aplikacija mogla postati još točnija i preciznija.

LITERATURA

- [1] *Android Developers*. Dohvaćeno iz <https://developer.android.com/guide/components/fragments> (13. 08 2019)
- [2] *Android Developers*. Dohvaćeno iz <https://developer.android.com/guide/topics/ui/layout/recyclerview> (12. 08 2019)
- [3] *Android Statistics..* Dohvaćeno iz <https://expandedramblings.com/index.php/android-statistics/> (15. 5 2019)
- [4] K. S. Bates, *Head First*. O'Reilly Media (2003).
- [5] *Fortune*. Dohvaćeno iz <https://fortune.com/2019/02/14/hacked-myfitnesspal-data-sale-dark-web-one-year-breach/> (21. 06 2019)
- [6] C. Gem, *Calorie Counter* (2013)
- [7] *Hrvatski zavod za javno zdravstvo* Dohvaćeno iz <https://www.hzjz.hr/sluzba-promicanje-zdravlja/odjel-za-prevenciju-debljine/> (28. 06 2019)
- [8] *Kadulja* Dohvaćeno iz <https://kadulja.com/zdravlje/sto-je-zapravo-zdrava-prehrana/> (15. 6 2019)
- [9] V. Lawrence, & P. Kopelman Medical consequences of obesity. *Clin Dermatol* , 296-302, (2004)
- [10] *Studentski*. Dohvaćeno iz <http://studentski.hr/studenti/zdravlje-i-ljepota/aplikacija-nutrimenza-pokazuje-kolicinu-kalorija-unesenih-obrokom-u-menzi> (18. 08 2019)
- [11] *Trgovina Play* Dohvaćeno iz <https://play.google.com/store/apps/details?id=com.fitnow.loseit> (06. 08 2019)
- [12] World Health Organization. Reducing risks, promoting healthy life. Geneva: World Health Organization; 2002.
- [13] A. Berghoefer, T. Pischon, T. Reinhold, C.M. Apovian, A.M. Sharma, S.N. Willich, Obesity prevalence from a European perspective: a systematic review (2008)

- [14] S. Milanović Demografske, bihevioralne i socio-ekonomske odrednice debljine u odraslih u Hrvatskoj. Sveučilište u Zagrebu, Medicinski fakultet, Zagreb, (2010)
- [15] L.V. Gall Managing obesity and diabetes. London: Science Press Ltd.; (2003)
- [16] B. Eckel: Thinking in Java (4th Edition), Prentice Hall; 4 edition (February 20, 2006)

ŽIVOTOPIS

Nikola Barbarić rođen je 13.11.1997 u Novoj Bili, Bosna i Hercegovina. Iako je rođen u Novoj Biloj sa prebivalištem u Žepču, gdje upisuje osnovnu školu OŠ ŽEPČE 2003. godine, te nakon odličnog uspjeha i osvojenog drugog mjesta na županijskom natjecanju iz matematike, upisuje srednju školu KŠC Don Bosco Žepče. Nakon što je završio srednju školu, 2016. godine upisao je preddiplomski sveučilišni studij Računarstvo na Fakultetu elektrotehnike, računarstva i informacijskih tehnologija u Osijeku. Android mu je zanimljiv, a poznaje programske jezike C, C++, Javu, HTML. Preferira rad u Javi, iako je počeo učiti Kotlin, koji smatra kvalitetnom zamjenom. Za sebe kaže da je ambiciozan i uporan.

SAŽETAK

Završni rad prati razvoj mobilne aplikacije za zdravu i kontroliranu prehranu. Ciljna grupa su studenti koji se hrane u studentskoj menzi, te su obuhvaćena sva jela koja menza ima u ponudi. Korisnik na osnovu svojih podataka, poput visine, godina, aktivnosti dolazi do podatka o preporučenom dnevnom unosu kalorija. Sva jela su mu razvrstana po kategorijama, unutar kojih je za svako jelo navedena kalorijska i nutritivna vrijednost. Odabirom određenih obroka, preporučeni unos se mijenja, te se prati količina kalorija za 24 sata. Svaki dan se kreće ponovno, dok se prethodni dani spremaju u online bazu podataka. U rješavanju je korišten Android Studio, opisni jezik XML i programski jezik Java. Aplikacija je prilagođena Android uređajima, jer su najrasprostranjeniji. Za elemente različitih aktivnosti su korišteni reciklirajući elementi poput fragmenata, recycler viewa ili tab layouta. Spremanje je postignuto pomoću SharedPreferences, dok se za online bazu podataka koristio FireBase. Ispitivanjem različitih slučajeva pokazalo se da aplikacija radi korisno i učinkovito, a tri su slučaja prikazana u završnom radu. Aplikacija bi se mogla unaprijediti uvođenjem dodatnih funkcionalnosti i detaljnijom teorijskom podlogom.

Ključne riječi: Android aplikacija, brojanje kalorija, debljanje, mršavljenje, student, zdrava prehrana

ABSTRACT

This paper follows the development of a mobile app for healthy and controlled eating. The target group is students who eat at the student canteen, and the app covers all the meals that the canteen has to offer. The app, based on users data, such as height, age, activity, comes up with the recommended daily calorie intake. All its dishes are classified into categories, where each calorie and nutritional value are indicated within. By selecting certain meals, the recommended intake changes and the calorie count is monitored for 24 hours. Every day it starts again, while the previous days are saved in an online database. The used solution was Android Studio, a descriptive XML language and a Java programming language. The app is tailored for Android devices because they are the most widespread. Recycling elements such as fragments, recycler views, or tab layouts were used for elements of various activities. Saving data was achieved by using SharedPreferences, while the online database used FireBase. Testing different cases has shown that the application works efficiently and effectively, and three cases are presented in this paper. The application could be enhanced with the introduction of additional functionality and a more detailed theoretical background.

Keywords: Android app, calorie counting, healthy food, weight gain, weight loss, student

PRILOZI

Prilog 1: Završni rad .docx forma

Prilog 2: Završni rad .pdf formata

Prilog 3: Android Studio projekat mobilne aplikacije