

Nutrition tracker

Dizajn, navigacija i organizacija ekrana

Napraviti aplikaciju za pretragu čuvanje i praćenje obroka kao i kreiranje planova ishrane.

Sve stavke su prepuštene vama, s tim što dizajn i komponente koje se koriste trebaju biti ispravno korišćene (material design specifikacija, korišćenje fragmenata, recycler view-ova itd.). Neispravno korišćenje view komponenti i loš dizajn će rezultovati oduzimanju poena.

Sadržaj

Nutrition tracker.....	1
Dizajn, navigacija i organizacija ekrana.....	1
Splash screen i Login screen (1p).....	3
Početna strana (3p).....	3
Strana za filtriranje (6p).....	3
Lista jela (4p).....	3
Tim.....	4
Detaljan prikaz jela (2p).....	4
Tim.....	4
Čuvanje jela (9p).....	4
Tim.....	4
Statistika (5p).....	5
Tim.....	5
Kreiranje plana (10p).....	5
Tim.....	5
Timski rad – calorie tracker.....	5
Tehnički detalji.....	6
Slanje projekta.....	6
Napomene o odbrani (30p).....	7
Detaljnije o bodovanju.....	7

Splash screen i Login screen (1p)

Na splash screen-u je potrebno proveriti da li se korisnik ranije prijavio, ukoliko jeste preusmeriti ga na glavni ekran, u suprotnom navesti ga na Login ekran.

Login ekran treba da sadrži polja za unos korisničkog imena i lozinke (neka je jedini uslov za lozinku da bude minimum dužine 4 karaktera, ostale limitacije su na vama).

Početna strana (3p)

Na početnoj strani se nalazi lista kategorija. Potrebno je da za svaku kategoriju bude prikazana njena slika i naziv u okviru liste kao i da postoji ikonica (na primer tri tačke sa desne strane stavke u listi) na koju kad se klikne prikaže se dijalog u kom piše opis kategorije. Pritiskom na neku od kategorija se otvara lista jela koja pripadaju toj kategoriji.

Na ovoj strani je takođe potrebno implementirati pretragu po nazivu sastojka ili jela bez da se odabere kategorija.

Strana za filtriranje (6p)

Na ovoj strani treba omogućiti pretragu po parametrima: kategorije, oblasti i sastojci. Filtriranje treba da radi tako što će se prvo odrediti parametar po kom će se filtrirati jela (na primer tabni meni), čime će se izlistati lista jela odabranog parametra (kategorije, oblasti ili sastojci). U okviru ove liste treba omogućiti sortiranje po abecedi, filtriranje po nazivu, filtriranje po tagovima i paginaciju (na primer da se prikazuje po 20 stavki u listi). Klikom na stavku (ili na dugme u okviru stavke, na vama je) se odlazi na prikaz odabranog jela.

NAPOMENA: Svaki put kada je u pitanju neka pretraga, potrebno je prikazati animaciju za učitavanje sve dok sveži podaci ne pristignu sa api-ja.

Lista jela (4p)

U okviru liste jela treba prikazati naziv jela i njegovu sliku. Na ovoj stranici je potrebno implementirati paginaciju tako da se vidi po 10 jela na strani. Pritiskom na neko od jela (ili dugmeta u okviru liste, na vama je) se otvara detaljan prikaz jela. Na stranici za listu jela je potrebno omogućiti da se vrši pretraga direktno preko naziva jela ili preko sastojka nevezano za to da li je prethodno odabrao kategoriju ili ih filtrirao po nekom drugom parametru.

Tim

Omogućiti filtriranje po broju kalorija jela (više, manje i između), kao i sortiranje po istim. Lista koja je dobijena po pretrazi preko kalorija treba u svakoj svojoj stavci da ima broj kalorija tog jela.

Detaljan prikaz jela (2p)

Na stranici za detaljan prikaz jela potrebno je prikazati naziv jela, kategoriju, oblast, instrukcije za pripremu, sliku, tagove, link ka snimku pripreme i sve sastojke sa merama. Na ovoj stranici treba da postoji opcija za čuvanje prikazanog jela.

Ukoliko neki od podataka (na primer link ka snimku pripreme) nije obezbeđen preko api-ja smao ga nemojte prikazati onda ili kao vrednost tog polja ostavite poruku (na primer *Not available*).

Tim

Prikazati i kalorijsku vrednost jela.

Čuvanje jela (9p)

Odabirom opcije za čuvanje jela, korisniku se otvara novi ekran gde se prikazuju naziv i slika odabranog jela i gde korisnik može da izabere datum kada je planirao da priprema ovaj obrok (podrazumevano se prikazuje današnji datum) i kategoriju obroka (doručak, ručak užina ili večera). Klikom na sliku jela, otvara se dijalog koji od korisnika traži potvrdu da li želimo da otvorimo kameru telefona (koristiti implicitni intent) i može napraviti novu fotografiju jela koja će zameniti podrazumevanu i nakon čuvanja će se koristiti. Nakon čuvanja jela korisniku se prikazuje obaveštenje da je jelo uspešno sačuvano u meni.

Korisnik je u mogućnosti da pregleda sačuvana jela i nakon svakog ponovnog pokretanja aplikacije tako da se jela za meni moraju čuvati u bazu podataka. Podatke koje je potrebno čuvati su naziv jela, slika jela (url ili putanja fajla), instrukcije za pripremu, link ka snimku pripreme, sastojci i mere, kategorija jela, datum kada je jelo planirano i za koji obrok je planirano.

Pretragu jela iz menija (ne sa api-ja) implementirati kao opciju koja se bira u okviru pretrage jela.

Takođe, potrebno je implementirati ekran za izmenu jela iz menija i opciju za brisanje jela iz menija.

Tim

Implementirati opciju za čuvanje kalorijske vrednosti jela. Ovo treba da radi tako što će dohvatiti kalorijsku vrednost svih sastojaka (koji imaju smisla, na primer listove bosiljka ne morate racunati) i u zavisnosti od gramaže u receptu tih sastojaka da sračuna kalorijsku vrednost celog jela. Ukoliko na api-ju nisu obezbeđeni podaci o svim sastojcima obavestiti korisnika o tome.

Takođe, treba implementirati opciju za čuvanje kalorijskih vrednosti sastojaka (na 100g) kako bi bilo moguće čuvati vrednosti jela na osnovu njih.

Statistika (5p)

Dodati stranicu koja će da sadrži statistiku o tome koliko je jela dodato za svaki dan u prethodnih 7 dana i prikazati ih u vidu grafa. Ove podatke treba vaditi iz lokalne baze.

Tim

Dodati opciju za menjanje između prethodno opisanog grafa i istog takvog grafa koji će se prikazivati statistiku po kalorijama umesto po broju obroka. Ukoliko broj kalorija po danu prelazi korisnikov dnevni unos (opisano kasnije) obojiti deo grafa u crveno, u suprotnom zeleno. Ukoliko za jela ne postoji kalorijska vrednost, obavestiti korisnika.

Kreiranje plana (10p)

Kreirati stranicu za pravljenje plana ishrane na nedeljnom nivou. Pri kreiranju plana se otvara forma za dodavanje jela po danima. Jela se mogu dodavati ili iz baze ili sa api-ja tako da se ovaj izbor mora omogućiti korisniku. Forma treba da bude pregledna i lako se navigirati kroz nju (prikaz dana, obroka itd...). Uokviru forme treba da se unese email kome treba poslati kreirani plan. Pri predavanju forme se šalje email (implicitnim intent-om). Uokviru mejla treba da se nalazi spisak jela uokviru plana, kao i njihov sadržaj (srediti body mail-a tako da bude pregledno) i link koji kad se klikne otvara aplikaciju.

Tim

Plan treba da sadrži i kalorijske vrednosti svih jela (ukoliko nema podataka o kalorijama za jela u planu nije ih moguće dodati bez da ove informacije nisu prethodno obezbeđene).

Timski rad – calorie tracker

Ukoliko odaberete opciju da radite projekat u paru, potrebno je kao dodatak koristiti API za praćenje kalorijske vrednosti jela. Ovaj API podržava praćenje kalorijskih vrednosti sastojaka

(sastojci se gledaju na 100g, tako da biste kalorijske vrednosti jela računali kao sabrane vrednosti kalorija sastojaka u odnosu na količinu grama tog sastojka koji ide u jelo, takodje vredosti koje su od manje vaznosti, tipa cenovi luka listovi bosiljka, ne moraju da se racunaju).

Takođe, treba da postoji način na koji će korisnik na osnovu svojih podataka (godine, visina, težina, pol, nivo aktivnosti) dobiti dnevni unos kalorija za održavanje kilaže (formule postoje online).

Napomena: ne možete odabrati da radite timsku opciju bez da implementirati deo sa kalorijama, odnosno da uradite timski opciju za individualan rad i da položite tako, neće se računati da je rađen timski rad.

Tehnički detalji

TheMealDB API: <https://www.themealdb.com/api.php>

Nutrition API: <https://api-ninjas.com/api/nutrition>

Kreiranje dijaloga za izbor datuma:

<https://developer.android.com/reference/android/widget/DatePicker>

<https://www.youtube.com/watch?v=qCoidM98zNk&list=PLguEs2eVqV9fOI9asSorWGh7PaYk0GmIY&index=2>

Kreiranje dijaloga za otvaranje kamere:

<https://developer.android.com/guide/topics/ui/dialogs#DialogFragment>

Obavezno je koristiti ViewModel-e i LiveData na prezentacionom sloju. LiveData bi trebalo umesto da direktno čuva strukture podataka, čuva stanja koja sadrže podatke. Svi podaci moraju biti dopremljeni iz repozitorijuma/sačuvani u repozitorijumu koji je u sloju podataka. Za rad sa bazom koristiti Room, a za komunikaciju sa serverom Retrofit.

Slanje projekta

Naziv projekta i paketa kada ga kreirate u Android studiju mora biti u formatu rs.rafi.projekat_jun_ime_prezime_brind.

Subject mail-a mora da bude u obliku: „[RMA 2023] P_JUN ime_prezime_ind“.

Npr. „[RMA 2023] P_JUN student_studentic_rn0101“.

Zipovan projekat nazvan mora biti ime_prezime_brind.zip i šaljete ga preko drive-a na sbudimac@raf.rs ili nredzic@raf.rs.

Pre zipovanja potrebno je uraditi clean project!

Pored ovoga možete poslati i link ka github repozitorijumu.

Ostali formati se neće priznavati!

Odbrana se drži na fakultetu, potrebno je da ponesete indeks sa sobom.

Napomene o odbrani (30p)

Odbrana nosi do 30 poena i smatra se kao usmeni deo ispita. Pitanja će biti praktične prirode i vezana za implementaciju projekta (na primer čemu služi i kako radi komponenta x, zašto ste je koristili umesto komponente y, šta će se desiti ako promenimo z, napravi promene tako da se dešava x umesto y itd.).

Detaljnije o bodovanju

Svaka celina nosi svoj broj poena naveden pored naziva. Podrazumeva se da je UI ispravno implementiran i da komunicira sa drugim komponentama (pokretanje activity-a, obrada rezultata, komunikacija preko deljenog viewmodel-a itd...). Bitan je vizualni izgled aplikacije (ispoštovana specifikacija dizajna, smislene margine, raspored elemenata, korišćenje adekvatnih view komponenti i sl.), u suprotnom će biti oduzimani bodovi.

Takođe je bitno ispoštovati arhitekturu, u suprotnom projekat (ili celine) vredi 1/3 poena.

Minimum poena na implementaciji je 20/40, na odbrani 15/30.