Aplikacija za priporočila receptov

Člane skupine: Nika Demšar (63210382), Urška Frelih Uhelj (63230067)

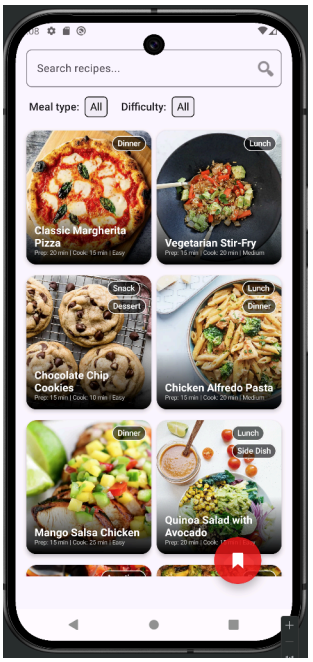
Opis: Najina aplikacija za recepte omogoča uporabnikom enostaven dostop do raznoraznih receptov. Vsebuje funkcionalnost iskanja, ki omogoča filtriranje receptov glede na ključne besede, ki jih uporabnik vpiše v vrstico za iskanje. Prav tako lahko s filtriranjem izbira med vrstamo obroka in težavnostjo priprave. Uporabnik lahko prav tako shrani priljubljene recepte za kasnejšo uporabo. S klikom na posamezen recept se prikaže podroben opis le-tega, ki vključuje sestavine, navodila za pripravo in druge pomembne informacije. Aplikacija je uporabniku prijazna in zelo enostavna za uporabo.

Uporabljene tehnologije: Android Studio, Kotlin, Jetpack Compose, Figma, Jetpack Navigation, SQLite, API

Github povezava: https://github.com/nirinja/EMP

Implementirane funkcionalnosti in primeri uporabe

KATALOG RECEPTOV



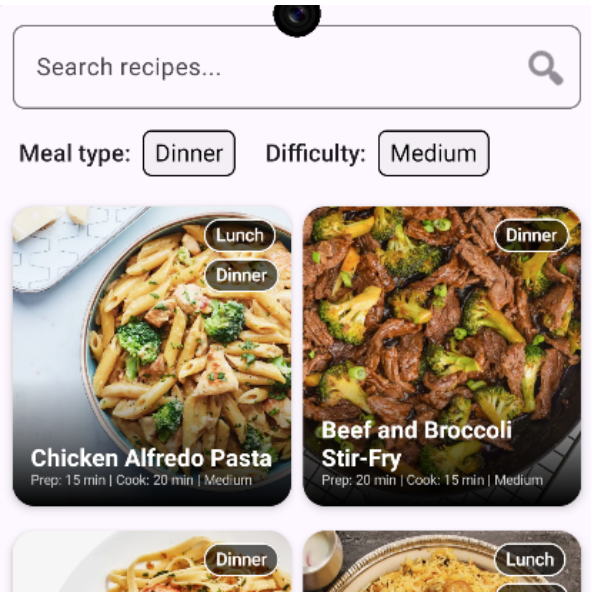
Glavna stran najine aplikacije prikazuje katalog izbranih receptov, med katerimi lahko uporabnik enostavno brska in najde idejo za naslednji obrok. Vsak posamezen recept je prikazan v svoji "kartici" pravokotne oblike. Vsaka kartica vsebuje naslov recepta, čas priprave in težavnost v levem spodnjem kotu in vrsto obroka v desnem zgornjem. Vsaka kartica je v spodnjem delu nekoliko zatemnjena, da je naslov recepta bolje viden.

Primer uporabe: že cel teden jem eno in isto hrano, zato si danes želim nekaj novega. Brskam po katalogu in najdem recept, ki mi najbolj pade v oči.

ISKANJE RECEPTOV

Na glavni strani se na vrhu nahaja orodna vrstica za iskanje. Uporabnik lahko vpiše ključno besedo in pritisne na ikono **"išči"** - lupo, ki se nahaja v zgornjem levem kotu. S tem bo najina aplikacija poiskala med vsemi recepti, ki so shranjeni v bazi podatkov (iz API) in jih prikazala na zaslon. Če uporabnik želi iskati

samo med recepti, ki so že prikazani, lahko samo napiše nekaj črk v isto orodno vrstico in takoj se bo aktiviralo iskanje.



Primer uporabe: za kosilo želim narediti pico, ampak se ne morem odločiti kakšne vrste naj bo. V orodno vrstico za iskanje vpišem ključno besedno: *Pizza* in pritisnem na gumb išči. Izberem med možnostmi, ki mi jih poda aplikacija.

FILTRIRANJE MED RECEPTI

Poleg iskanja se na glavni strani, pod orodno vrstico za iskanje nahajata tdi dva filtra. Prvi omogoča filtriranje med vrstami obrokov, drugi pa med težavnostmi.

Primer uporabe: sem še kuharica začetnica in rada bi za večerjo svojemu fantu pripravila presenečenje. Med filtri izberem vrsto obroka: Večerja in težavnost: Preprosto in izberem med podanimi recepti.

PODSTRAN - PODROBNOSTI O RECEPTU

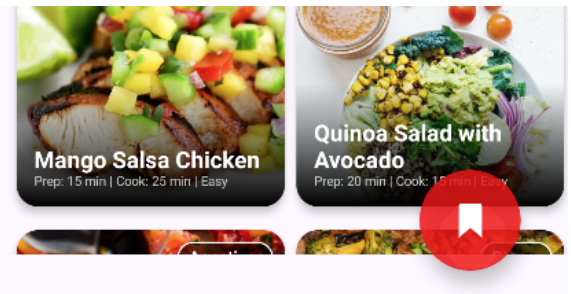
Ob pritisku na katerikoli recept iz kataloga se mi odpre nov zaslon, kjer so o željenem receptu napisane vse pomembne informacije. Najprej so vidni naslov recepta, čas predpriprave, čas kuhanja in zahtevnost. Sledijo sestavine, ki jih potrebujem za izvedbo in na dnu so navodila, korak po koraku.

V desnem zgornjem kotu se nahaja še gumb za shranjevanje recepta v zbirko svojih shranjenih receptov, v levem zgornjem kotu pa je možnost za vrnitev na glavni zaslon z gumbom **Nazaj**. Slika recepta na tej strani je ponovno nekoliko zatemnjena iz zgornje strani, da sta oba gumba dobro vidna na katerikoli podlagi.

Primer uporabe: želim narediti tiramisu a pojma nimam kakšen je postopek. Pritisnem na recept in si preberem vse potrebne informacije. Končni rezultat je odličen, saj sem imela pripravljene vse sestavine in razumela postopek.

SHRANJEVANJE RECEPTOV

V spodnjem desnem kotu se na glavni strani nahaja gumb za dostop do zbirke shranjenih receptov. Recepte lahko vanjo dodajamo tako, da jih s pritiskom na gumb v desnem zgornjem kotu podstrani za podrobnosti o receptu, dodamo v zbirko. Gumb shrani se obarva in tako vemo, da je recept varno pospravljen. Če smo se zmotili in recepta ne želimo med shranjenimi recepti, lahko ponovno pritisnemo

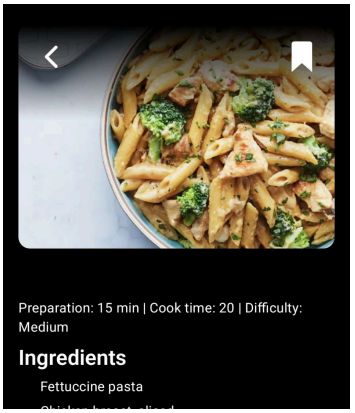


na isti gumb, ikona ne bo več obarvana in recept bo izginil iz zbirke shranjenih.

Primer uporabe: Na avtobusu imam nekaj časa, zato brskam po katalogu receptov. Najdem recept, ki bi ga rada pripravila danes za večerjo, zato ga preprosto shranim v svojo zbirko. Zvečer ponovno odprem aplikacijo, se pomaknem med svoje shranjene recepte, kjer me že čakajo navodila za današnjo večerjo, hitro in brez iskanja.

Arhitekturna zasnove aplikacije

Aplikacija ima centralno MainActivity, ki naloži in prikaže seznam receptov s pomočjo Jetpack Compose. Za podatkovno plast je uporabljena Room baza, ki shranjuje podatke o receptih, vključno z njihovimi kategorijami.
RecipeActivity je odgovorna za prikaz podrobnosti posameznega recepta. Recept se prenese prek Intent objekta kot JSON podatek. Aktivnost omogoča uporabniku, da shrani ali odstrani recept iz baze z uporabo DAO metod (insertRecipe, deleteRecipeByName). Logika za preverjanje, ali je recept že shranjen, se izvaja z uporabo Room poizvedb v kombinaciji z LaunchedEffect v Compose-u, kar omogoča samodejne posodobitve stanja (ikone za shranjevanje).
Uporabniški vmesnik je odziven, zlahka berljiv in omogoča učinkovito upravljanje receptov, medtem ko so podatki in logika obdelave podatkov združeni znotraj aktivnosti.



Vgrajen pa imava tudi senzor svetlobe, katerega uporabi za to, da spremeni temo aplikacije grede na svetlost, Svetlo avtomatsko se uporabi svetlo temo in obratno.

Demonstracija operacij ob različnih časih v življenjskem ciklu aktivnosti

Življenjski cikel aktivnosti vključuje metode, ki se izvajajo ob določenih dogodkih. onCreate() se kliče, ko se aktivnost prvič ustvari, in je primeren za inicializacijo uporabniškega vmesnika in podatkov. onStart() nastopi, ko aktivnost postane vidna, kar omogoča pripravo virov za prikaz uporabniku. onResume() pomeni, da je aktivnost interaktivna, idealna za začetek opazovanja podatkov v realnem času. onPause() se uporablja za shranjevanje trenutnih nastavitev ali ustavitev nepotrebnih operacij, ko je aktivnost le v ozadju. onStop() sprosti vire, kot so senzorji ali omrežne povezave, ker aktivnost ni več vidna. onRestart() omogoča pripravo pred ponovnim zagonom aktivnosti. Na koncu onDestroy() sprosti vse vire, ko je aktivnost popolnoma odstranjena.

OnResume in onPause sva tudi uporabili pri zaznavanju svetlobe in spreminjanje tem.

Uporaba ViewModel, UI State in StateFlow

MVVM (Model, View, ViewModel) omogoča shranjevanje in upravljanje UI-podatkov neodvisno od življenjskega cikla aktivnosti. Omogoča odpornost na spremembe, kot je rotacija zaslona, in samodejno posodablja UI ob spremembah podatkov, kar izboljša modularnost in odzivnost aplikacije.

UI State predstavlja trenutno stanje uporabniškega vmesnika in se v Jetpack Compose upravlja z remember ter mutableStateOf, kar omogoča takojšnje posodobitve vmesnika. To sva tudi uporabile pri shranjevanju receptna na podatkovno bazo.

Za učinkovito posodabljanje podatkov v realnem času se uporablja StateFlow, ki sinhronizira podatke med ViewModel in UI ter omogoča samodejne posodobitve ob spremembah v bazi.

Uporaba zunanjih API-jev v vaši aplikaciji

V najini aplikaciji sva uporabili zunanji API, ki sva ga pridobili na povezavi:

DummyJSON - https://dummyjson.com/docs/recipes

Ta API zajema vse potrebne informacije, ki sva jih potrebovali za sestavo najine aplikacije:


```
{
  "recipes": [
    {
      "id": 1,
      "name": "Classic Margherita Pizza",
      "ingredients": [
        "Pizza dough",
        "Tomato sauce",
        "Fresh mozzarella cheese",
        "Fresh basil leaves",
        "Olive oil",
        "Salt and pepper to taste"
      ],
      "instructions": [
        "Preheat the oven to 475°F (245°C).",
        "Roll out the pizza dough and spread tomato sauce evenly.",
        "Top with slices of fresh mozzarella and fresh basil leaves.",
        "Drizzle with olive oil and season with salt and pepper.",
        "Bake in the preheated oven for 12-15 minutes or until the crust is golden brown.",
        "Slice and serve hot."
      ],
      "prepTimeMinutes": 20,
      "cookTimeMinutes": 15,
      "servings": 4,
      "difficulty": "Easy",
      "cuisine": "Italian",
      "caloriesPerServing": 300,
      "tags": [
        "Pizza",
        "Italian"
      ],
      "userId": 166,
      "image": "https://cdn.dummyjson.com/recipe-images/1.webp",
      "rating": 4.6,
      "reviewCount": 98,
      "mealType": [
        "Dinner"
      ]
    },
```


Shema in opis podatkovnega modela

Naš podatkovni model vsebuje vsebuje dve povezani tabeli: recipes (recepti) in categories (kategorije).

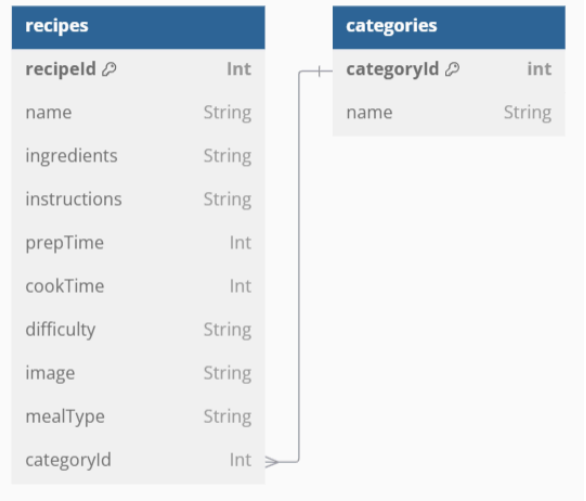


Tabela recipes shranjuje podrobnosti o shranjenem receptu, vključno z njegovim unikatnim recipeId, imenom, sestavinami, navodili, časom priprave in kuhanja, težavnostjo, povezavo do slike, vrstami obrokov ter categoryId, ki je zunanji ključ za tabelo categories.

Tabela categories vsebuje categoryId, ki služi kot primarni ključ, in ime kategorije, kot so npr. "Sladice" ali "Glavne jedi". Relacija med tabelama je ena proti mnogim, kar pomeni, da lahko ena kategorija vključuje več receptov, medtem ko vsak recept pripada eni kategoriji. Takšna struktura omogoča enostavno razvrščanje receptov glede na kategorije in hitro pridobivanje podatkov za aplikacijo.