DOKUMENTACIJA

* Opis rada aplikacije

Aplikacija ShopEase je moderna Android aplikacija razvijena za pregled i kupovinu različitih proizvoda putem integracije s Web API-jem. Cilj aplikacije je pružiti korisnicima intuitivno iskustvo pri pregledu i kupovini proizvoda iz različitih kategorija.

* Početni ekran (Landing page)

Na početnom ekranu korisnici mogu pregledati različite kategorije proizvoda poput elektronike, nakita, odjeće za muškarce i žene. Svaka kategorija prikazana je kao kartica s odgovarajućom slikom i nazivom.

Elementi početnog ekrana:

1. Top traka (TopAppBar):

* Prikazuje naziv aplikacije i opciju za navigaciju unatrag (back) ili otvaranje košarice.
* Omogućuje navigaciju na druge dijelove aplikacije, poput ekranu za kupovinu ili povratak na početni ekran.

1. Kategorije proizvoda:

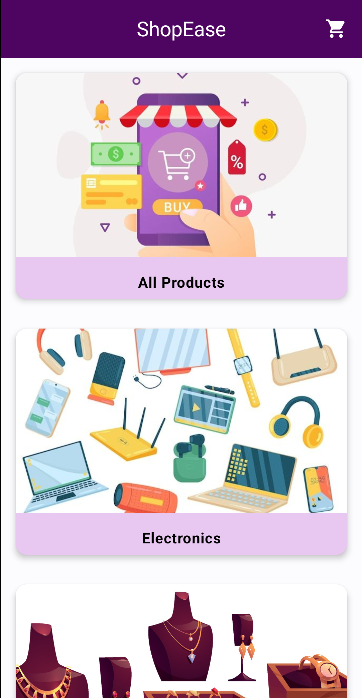
* Prikazuju se kao kartice (CategoryCard) koje sadrže sliku i naziv kategorije proizvoda.
* Korisnik može kliknuti na svaku karticu kako bi vidio proizvode u toj kategoriji.

1. Klikabilne kartice (CategoryCard):

* Svaka kartica sadrži sliku koja predstavlja kategoriju proizvoda.
* Ispod slike nalazi se naziv kategorije.
* Korisnik može kliknuti na karticu kako bi prikazao proizvode u toj kategoriji.

1. Optimizacija za različite orijentacije:

* Aplikacija je prilagođena za vertikalni i horizontalni prikaz ekrana.
* Prikaz kategorija proizvoda je optimiziran za najbolje iskustvo korisnika, bez obzira na orijentaciju ekrana.



* Ekran sa artiklima

"CategoryScreen" je ključni ekran aplikacije ShopEase koji korisnicima omogućuje pregled proizvoda u odabranim kategorijama. Sadrži top traku s nazivom kategorije i navigacijskim dugmićima te prikazuje dinamički sadržaj ovisno o stanju: prikazuje sliku tokom učitavanja, poruku o greški u slučaju problema, ili listu artikala (proizvoda) u karticama za uspješno dohvaćene podatke. Svaka kartica artkila omogućuje klik za detaljnije informacije, čineći pregled proizvoda intuitivnim i prilagođenim različitim korisničkim iskustvima.

* Elementi ekrana sa artiklima

1. Top traka (TopAppBar):

* Prikazuje naziv kategorije proizvoda kao naslov.
* Sadrži dugmić za povratak na prethodni ekran (ako je moguće) i gumb za otvaranje košarice.

1. Glavni sadržaj:

* Ovisno o stanju (Loading, Success ili Error), prikazuje se odgovarajući ekran:
  + **LoadingScreen**: Prikazuje se slika koja označava da se učitavaju podaci.
  + **ErrorScreen**: Prikazuje se poruka o grešci i ikona za grešku. Omogućuje ponovno pokušavanje.
  + **ResultScreen**: Prikazuje listu artikala (proizvoda) u obliku kartica.

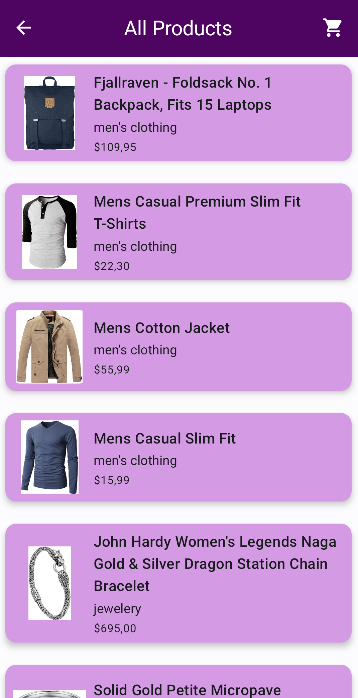
1. Kartice članaka (ArticleItem):

* Svaki artikal (proizvod) prikazan je unutar kartice.
* Kartica ima sliku artikla, naslov, kategoriju i cijenu.
* Omogućuje klik na artikal kako bi se otvorili detalji tog artikla.

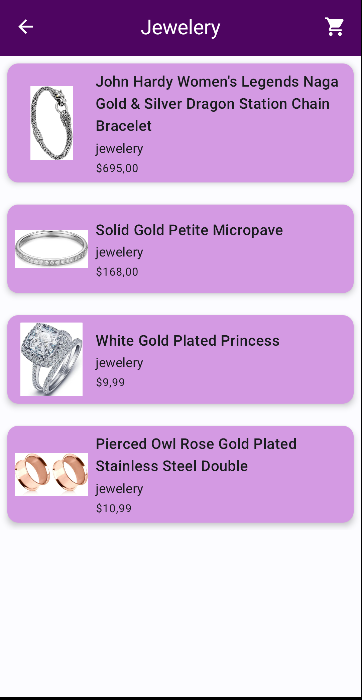
1. Optimizacija za različita stanja:

* Ekran je optimiziran za prikaz tokom učitavanja (Loading), uspješnog dohvaćanja podataka (Success) ili prikazivanja greške (Error).
* Prilagođava se kako bi korisnicima omogućio što jednostavnije i intuitivnije iskustvo pregledavanja proizvoda.

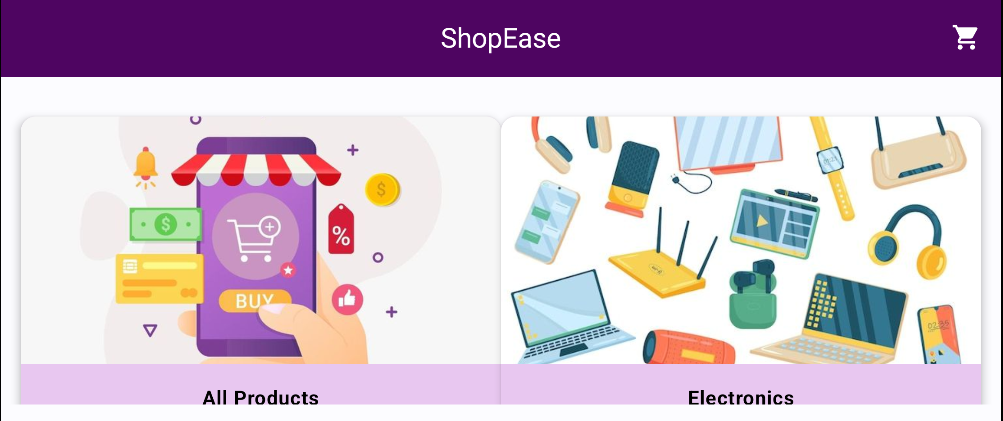
All products kategorija:



Jewelery kategorija:



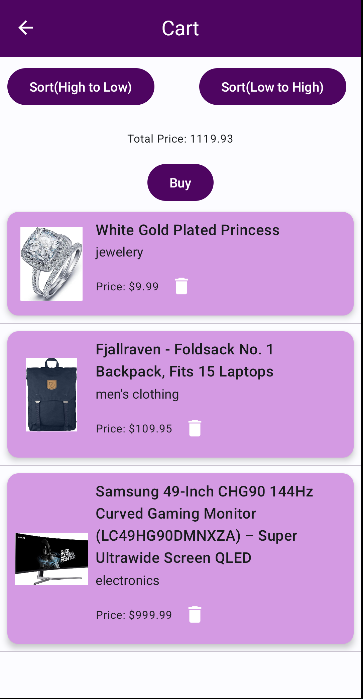
Aplikacija posjeduje pejzažni i portretni prikaz ekrana

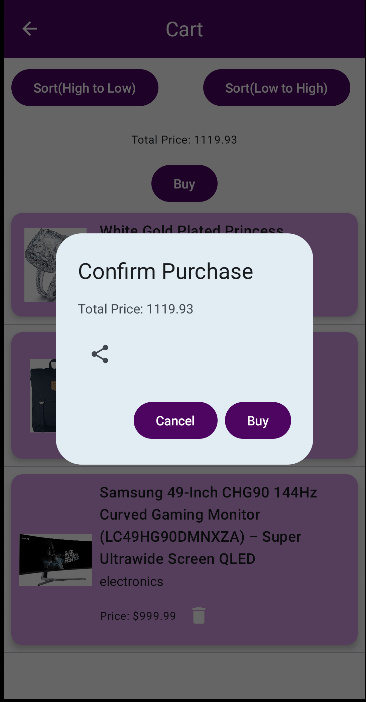
Kada je korisnik u portret mode-u svi sve kartice sa kategorijama artikala su poredane u jednoj koloni jedni ispod drugih (LazyColumn). S druge strane kada je korisnik u pejzažnom mode-u kategorije proizvoda su poredane u dvije kolone (LazyVerticalgrid) 

Ovi načini rada prilagođavaju se orijentaciji uređaja i omogućuju korisnicima optimalan prikaz kategorija proizvoda ovisno o veličini ekrana i korisničkom iskustvu.

* Ekran sa košaricom

Ekran sa košaricom je ekran na kojem kupci mogu vidjeti šta su do sada odlučili kupiti. Sadrži dva dugma za sortiranje elemenata po cijeni, tekst u kojem je prikazana ukupna cijena artikala u košarici, dugme za kupovinu I prikaz svih artikala u vidu Card komponenata. Ukoliko se kupac predomisli I više ne želi kupiti određeni artikal može pritisnuti na delete Icon koji se nalazi na svakoj Card komponenti I time će se artikal ukloniti iz košarice. Kada je kupac zadovoljan artklima može kliknuti na dugme Buy kada će se pojaviti AlertDialog gdje će biti prikazana ukupna cijena, dugme za share-ovanje artikala koje kupac želi kupiti kao I Cancel I Buy dugma za odustajanje od kupovine ili potvrdu kupovine. Kada se klikne Buy dugme svi artkili se brišu iz baze podataka I vraća se na Landing page





* Ekran sa detaljima artikla

Ovaj ekran predstavlja "ArticleDetailsScreen" i prikazuje detalje o pojedinačnom članku. Korisnici mogu vidjeti sliku, naslov, cijenu, opis, kategoriju i ocjene artikla. Pored toga, omogućava dodavanje artikla u korpu putem plutajućeg dugmeta, a na dnu ekrana se pojavljuje prilagođena snackbar obavijest koja potvrđuje da je artikal dodan u korpu.

* Elementi ekrana i njihov opis:

1. Plutajuće dugme (FloatingActionButton):
   * Dugme za dodavanje artikla u korpu.
   * Prikazuje snackbar obavijest kada je artikal dodan u korpu.
2. SnackbarHost:
   * Prilagođena snackbar obavijest koja se prikazuje kada je artikal dodan u korpu.
   * Sadrži ikonu potvrde (zelena kvačica) i poruku.
3. Slika artikla (AsyncImage):
   * Prikazuje sliku artikla.
   * Postavljena je da zauzima 300 dp visine i cijelu širinu ekrana.
   * Sadrži efekte prelaska (crossfade).
4. Prilagođena Snackbar (CustomSnackbar):
   * Prikazuje prilagođenu snackbar obavijest.

* **Opis arhitekture aplikacije**

Korištena je MVVM (Model-View-ViewModel) arhitektura i Jetpack Compose biblioteke za izradu korisničkog interfejsa.

* Slojevi arhitekture:

1. View (Pogled)
2. ViewModel
3. Model

* View

View sloj sadrži sve UI komponente koje su izgrađene koristeći Jetpack Compose. Ovaj sloj je odgovoran za prikazivanje podataka i prikupljanje korisničkih interakcija. Ekrani koji spadaju u ovaj sloj su BuyScreen, HomeScreen, CategoryScreen, ArticleDetailsScreen

* ViewModel

ViewModel sloj je odgovoran za upravljanje logikom aplikacije i poslovnim podacima. ViewModel komunicira sa Model slojem za dohvaćanje podataka i s View slojem za ažuriranje UI-ja. U aplikaciji ShopEase, ViewModel uključuje:

* ArticleViewModel**:** Ovaj viewModel se koristi za upravljanje artiklima u aplikaciji, uključujući dohvaćanje, sortiranje, dodavanje i brisanje artikala.
* HomeViewModel: Ovaj viewModel se koristi za upravljanje stanjem prikaza početnog zaslona aplikacije. U njemu se prikupljaju i prikazuju artikli iz različitih kategorija.
* Model

Model sloj predstavlja poslovnu logiku i podatke aplikacije. U ovom slučaju, model sloj uključuje:

* **data paket**:
  + **ArticleDao**: DAO (Data Access Object) za pristup bazi podataka.
  + **ArticleDatabase**: Konfiguracija baze podataka.
  + **ArticleItem**: Klasa koja predstavlja stavku artikla.
  + **ArticleRepository**: Repozitorij koji pruža apstrakciju za pristup podacima.
* **model paket**:
  + **Article.kt**: Klasa koja definira model artikla.

### Ostalo

* **network paket**:
  + **StoreApiService.kt**: Definira API servis za mrežne operacije, kao što su dohvaćanje podataka o artiklima.
* **navigation paket**:
  + **NavigationDestination.kt**: Interface koji definira navigacijske destinacije.
  + **NavGraph.kt**: Definira navigacijsku grafu aplikacije.
* **Opis funkcionalnosti pojedinačnih klasa**

#### 1. data Paket

* **ArticleDao**:
  + Definira metode za pristup bazi podataka artikala. Obično uključuje metode za umetanje, brisanje, ažuriranje i dohvaćanje artikala.
* **ArticleDatabase**:
  + Konfigurira bazu podataka koristeći Room ORM. Ovdje se obično definiraju entiteti i verzija baze podataka.
* **ArticleItem**:
  + Predstavlja entitet artikla u bazi podataka. Sadrži atribute kao što su naslov, cijena, slika, kategorija i ocjena artikla.
* **ArticleRepository**:
  + Služi kao apstrakcijski sloj između ViewModel-a i DAO-a. Pruža metode za dohvaćanje, dodavanje, brisanje i ažuriranje artikala te za dohvaćanje cijene svih artikala.

#### 2. model Paket

* **Article**:
  + Definira podatkovni model artikla koji se koristi u aplikaciji. Obično sadrži atribute kao što su naslov, cijena, opis, slika, kategorija i ocjena.

### ViewModel Sloj

#### 1. ui.theme Paket

* **ArticleViewModel**:
  + Upravlja podacima i poslovnom logikom vezanom za artikle. Sadrži metode za dohvaćanje artikala, dodavanje artikala u košaricu, brisanje artikala i ažuriranje ukupne cijene.
* **ArticleViewModelFactory**:
  + Pruža instancu ArticleViewModel s potrebnim ovisnostima, poput ArticleRepository.
* **HomeViewModel**:
  + Upravlja podacima za početni zaslon aplikacije. Sadrži metode za dohvaćanje artikala po kategorijama i upravljanje stanjem korisničkog sučelja (učitavanje, uspjeh, greška).

### View Sloj

#### 1. ui.theme Paket

* **ArticleDetailsScreen.kt**:
  + Prikazuje detalje odabranog artikla. Omogućava dodavanje artikla u košaricu i prikazuje obavijest o uspjehu dodavanja.
* **BuyScreen.kt**:
  + Prikazuje zaslon za kupovinu, obično uključuje prikaz artikala u košarici i ukupnu cijenu.
* **CategoryScreen.kt**:
  + Prikazuje različite kategorije artikala koje korisnik može pregledavati.
* **HomeScreen.kt**:
  + Prikazuje početni zaslon aplikacije, obično uključuje popis artikala ili kategorija za pregledavanje.
* **MainActivity**:
  + Glavna aktivnost koja pokreće aplikaciju i postavlja početnu navigaciju.
* **NavGraph.kt**:
  + Definira navigacijsku grafu aplikacije, uključujući sve moguće rute i destinacije unutar aplikacije.

### Ostalo

#### 1. network Paket

* **StoreApiService.kt**:
  + Definira API servis za mrežne operacije. Sadrži metode za dohvaćanje podataka o artiklima iz udaljenog izvora (API-a).

**ArticleDao**:

* **Funkcije**:
  + insertItem(article: ArticleItem): Umeće novi artikal u bazu podataka.
  + deleteItem(article: ArticleItem): Briše određeni artikal iz baze podataka.
  + getAllItems(): Dohvaća sve artikle iz baze podataka.
  + getAllItemsAsc(): Dohvaća sve artikle iz baze podataka sortirane po cijeni uzlazno.
  + getAllItemsDesc(): Dohvaća sve artikle iz baze podataka sortirane po cijeni silazno.
  + getPrice(): Dohvaća ukupnu cijenu svih artikala u bazi podataka.
* **Opis opštih koncepata Android frameworka**

Opšti koncepti Android frameworka obuhvataju osnovne principe i komponente koje čine strukturu Android operativnog sistema i omogućavaju razvoj mobilnih aplikacija.

 **Activities (Aktivnosti)**: Koriste se za predstavljanje korisničkog interfejsa na ekranu i za interakciju sa korisnikom. U projektu su korištene aktivnosti poput BuyScreen i HomeScreen kako bi se prikazale odgovarajuće stranice aplikacije.

 **ViewModel**: ViewModel je arhitekturni obrazac koji se koristi za čuvanje i upravljanje podacima koji su povezani sa UI komponentama. HomeViewModel u projektu čuva logiku i stanje igre.

 **Layouts (Rasporedi)**: Layouti definišu strukturu i izgled korisničkog interfejsa, uključujući poziciju i stil elemenata kao što su dugmad, polja za unos teksta, i slično.

 **Intents**: koriste se za pokretanje komponenti aplikacije, slanje i primanje podataka između komponenti, kao i za pokretanje aktivnosti iz različitih dijelova aplikacije. U projektu se koristi Intent za dijeljenje artikala putem drugih aplikacija.

 **Navigation (Navigacija)**: Navigacija se koristi za upravljanje prelaskom između različitih ekrana ili dijelova aplikacije. U projektu se koristi NavHost sa definisanim rutama (composable) kako bi se omogućilo navigiranje između početne stranice (HomeScreen) i ostalih stranica (CategoryScreen npr.).

* Localization: Proces prilagođavanja aplikacije različitim jezicima, regionalnim podešavanjima i kulturnim kontekstima kako bi se korisnicima pružilo lokalizovano iskustvo. Uključuje prevođenje tekstova, prilagođavanje formata datuma, vremena i valuta, kao i prilagođavanje drugih aspekata aplikacije prema jezičkim preferencama korisnika.
* Životni ciklus (Lifecycle): Odnosi se na seriju stanja koje aktivnost ili fragment može prolaziti tokom svog postojanja, od stvaranja do uništenja. Ovo je ključni koncept jer omogućava android programerima da upravljaju ponašanjem aplikacije u različitim situacijama, kao što su promjene konfiguracije uređaja, prelazak između aktivnosti i fragmenta, kao i promjene u životnom ciklusu samog Android sistema.
* Room

Room je objektno-relacijski okvir (ORM) za Android koji pruža apstrakcijski sloj preko SQLite baze podataka, omogućujući jednostavnije i sigurnije upravljanje lokalnim podacima u Android aplikacijama. Osnovni koncepti i komponente Room-a uključuju:

### Komponente Room-a

1. **Entity**:
   * Entitet predstavlja tablicu u bazi podataka. Svaka instanca entiteta odgovara retku u tablici.
   * Definira se pomoću @Entity anotacije.
2. **DAO (Data Access Object)**:

* DAO je sučelje koje sadrži metode za pristup bazi podataka. Koristi anotacije za definiranje SQL upita.
* Definira se pomoću @Dao anotacije.

1. **Database**:

* Klasa baze podataka predstavlja SQLite bazu podataka i djeluje kao glavna pristupna tačka za vezu s podacima.
* Definira se pomoću @Database anotacije.
* API

API (Application Programming Interface) je skup definicija i protokola koji omogućuju različitim softverskim komponentama da međusobno komuniciraju. U kontekstu weba, API omogućuje klijentskim aplikacijama da komuniciraju s poslužiteljskim aplikacijama putem HTTP zahtjeva.

Retrofit je popularna biblioteka za Android koja olakšava rad s API-ima. Koristi se za stvaranje HTTP zahtjeva i upravljanje odgovorima na vrlo jednostavan način.

### Komponente i funkcionalnosti koje ste definirali

#### 1. BASE\_URL

* BASE\_URL definira osnovnu URL adresu API-ja. U ovom slučaju, to je https://fakestoreapi.com. Svi HTTP zahtjevi koje aplikacija šalje bit će upućeni na ovu baznu adresu.

#### 2. Retrofit.Builder

* Retrofit.Builder je graditelj za stvaranje Retrofit instance. Ovdje se konfigurira Retrofit objekt pomoću addConverterFactory i baseUrl metoda.
* addConverterFactory koristi Json.asConverterFactory za dodavanje konvertera koji omogućuje serijalizaciju i deserializaciju JSON objekata koristeći kotlinx.serialization.
* baseUrl postavlja osnovnu URL adresu za sve HTTP zahtjeve.

Repzitorijum služi kao posrednik između aplikacije i baze podataka. On pruža apstrakciju preko operacija na podacima koje omogućavaju jednostavniji pristup i upravljanje podacima. Ključne prednosti korištenja repozitorija su:

1. **Apstrakcija Sloja za Pristup Podacima**: Repozitorijum izdvaja sloj za pristup podacima (DAO) iz logike aplikacije, čime se omogućava čistiji i modularniji kod.
2. **Jednostavnije Testiranje**: Koristeći repozitorijum, lakše je zameniti konkretne implementacije tokom testiranja, omogućavajući pisanje testova koji ne zavise od stvarne baze podataka.
3. **Centralizacija Poslovne Logike**: Logika vezana za manipulaciju podacima može biti centralizovana unutar repozitorijuma, što čini kod lakšim za održavanje.

* Zaključak

ShopEase pruža intuitivan i atraktivan korisnički interfejs koji olakšava korisnicima istraživanje i kupovinu različitih kategorija proizvoda. Korisnici mogu brzo navigirati između različitih kategorija poput svih proizvoda, elektronike, nakita te odjeće za muškarce i žene. Svaka kategorija predstavljena je kroz privlačnu karticu sa slikom i nazivom proizvoda, što omogućava korisnicima da brzo prepoznaju željenu kategoriju.

Osim estetskog aspekta, aplikacija nudi i funkcionalnosti kao što su mogućnost direktnog pristupa detaljima proizvoda i njihovoj kupovini s jednim klikom. Svaka kategorija povezana je s odgovarajućim funkcionalnostima koje korisnicima omogućavaju da istraže više ili dodaju proizvode u košaricu za kupovinu.

Dodatno, prilagodljivost aplikacije na različite orijentacije ekrana omogućava korisnicima ugodno iskustvo bez obzira na to koriste li uređaj u portretnom ili pejzažnom načinu.

Ukupno gledano, ShopEase čini online kupovinu jednostavnom i privlačnom aktivnošću, pružajući korisnicima sve što im je potrebno za pregled, odabir i kupovinu proizvoda iz udobnosti njihovog doma ili bilo gdje drugdje.