

Emnekode: (PGR208)

Emnenavn: (Android Programmering)

Antall ord: 937

(prosjektoppgave)

Høyskolen Kristiania



Semester: Høst 2023

Denne oppgaven er gjennomført som en del av utdannelsen ved Høyskolen Kristiania. Høyskolen er ikke ansvarlig for oppgavens metoder, resultater, konklusjoner eller anbefalinger.

PGR208 Eksamen – rapport

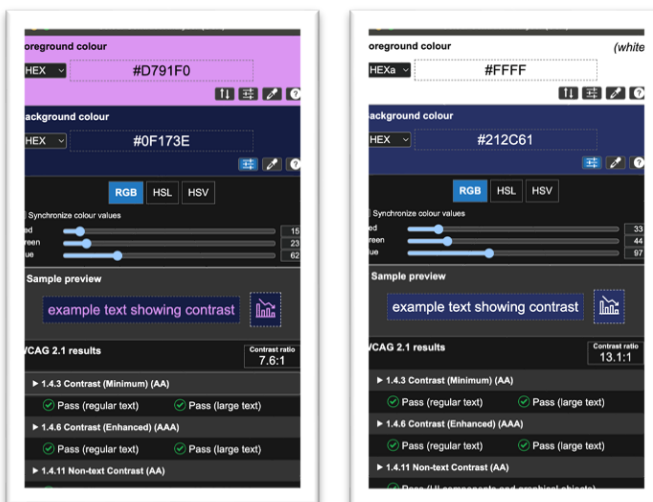
I denne oppgaven har vi designet og utviklet en fungerende shopping applikasjon ved hjelp av Android Studio. For å få til en funksjonell og helhetlig app, har vi tatt i bruk ulike teknikker og rammeverk, inkludert Jetpack Compose, Room for lokal lagring, API-integrasjon, og ViewModel.

Brukergrensesnitt

Når det gjelder brukergrensesnitt, har vi forsikret oss om at designet er enkelt og forståelig, slik at det blir en brukervennlig opplevelse for brukerne. Vi har tatt i bruk en enkelt layout som skal være innbydende, samtidig som den gir informasjon om hva de ulike elementene er. I tillegg har vi passet på at designet er konsistent gjennom alle de ulike skjermene, og at ikoner og lignende også har en relasjon, så det er en tydelig sammenheng.

Tilgjengelighet

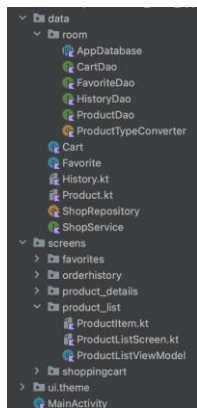
Vi har vært bevisste på ulike tilgjengelighets aspekter, som farge-kontrast og nedsatt synsevne. Dette har vi tatt hensyn til og implementert i designet vårt ved å sjekke at bakgrunns og tekst fargene vi har valgt består WCAG kontrast sjekken, slik at teksten er synlig og lesbar. I tillegg har vi satt beskrivende tekst under ikonene i navigasjonen og satte ved “contentDescription” i koden til bildene.



(Screenshot fra Colour Contrast Analyser app)

Kodestruktur

Når det gjelder kodestruktur, har vi forsøkt å opprettholde en modulær og lett leselig og forståelig kode. Vi er klar over at koden vår kunne ha blitt enda mer modulisert, grunnet noe redundant kode ulike steder, men vi føler vi har relativt god orden og oversikt over koden vår. Vi har organisert koden i logiske og relaterte klasser og pakker, slik at det er enkelt å finne frem. I tillegg har vi gitt de ulike filene, funksjonene, variablene etc. fornuftige og beskrivende navn, slik at det er enklere å skjønne koden og konteksten.



Vi har forsøkt å sette opp en enkel og organisert struktur på de ulike mappene, data--> room og screens med mapper igjen til de ulike skjermene og viewmodellen som hører til. Vi har også kommentert i koden vår for å organisere og tilføye ekstra informasjon som kan gjøre det enklere å skjønne koden vår.

Rammeverk

De ulike rammeverkene vi har tatt i bruk for å kunne utvikle denne shopping appen, er blant annet Jetpack Compose, som vi tok i bruk for å opprette en effektiv måte å navigere mellom de ulike skjermene i appen. Vi har også tatt i bruk local storage med Room som legger til rette en sikker måte å lagre lokale data på enheten, slik at vi kan få tilgang til produktene i apiet, selv uten internett.

Ved hjelp av Retrofit har vi gjennomført HTTP-forespørsler for å hente og integrere Fakestore API-et (<https://fakestoreapi.com/>). Dette har gjort det mulig for oss å innhente relevant produktinformasjon, som er nødvendig for å hente og vise de ulike produktene og info som følger med. I tillegg har vi brukt ViewModel for å strukturere og kunne sette opp de nødvendige funksjonene og tilstandene, slik at ulike endringer og tilstander kan gjennomføres, oppdateres og bli tatt vare på.

Refleksjon

- Hva ville vi ha gjort annerledes?

Til neste gang ville vi ha satt mer tid på forarbeidet, som å notere ned alle funksjoner vi trenger å bruke, og også få satt opp data og room før vi startet på UI.

- *Hva brukte vi mye tid på?*

Brukte mye tid på å konvertere riktig fra data type til data type. Det var vanskelig å forstå hvordan man skulle sette opp konverteringene når det kom til bildet med tanke på noen APIer hadde image som en string med objekter inni et objekt med string. Brukte mye tid på å få frem APIet, prøvde oss frem med 3 forskjellige APIer. Fikk til å konvertere riktig fra et objekt av et array med image som en string, kunne vi endelig vise produkter fram på skjermen. I tillegg slet vi mye med å opprettholde koblingen med apiet når vi først hadde den. På starten ble det ofte brutt, og vi endte opp med å måtte bytte mellom ulike database versjoner inne i room. Koblingen mellom produkt historie og handle kurven var vanskelig å sette opp. Klarte til slutt å få lagt til produkter gjennom kjøps historien til brukeren, men dette kun i loggen. Var vanskelig å forstå hvordan å få handle kurven til å oppdatere seg uten at appen stopper

Til slutt slet vi med at shoppingCartScreen ikke ville oppdatere seg like fort når et produkt ble lagt i handlekurven. Vi har ennå ikke fått det helt til, så noen ganger dukker produktet opp med en gang, andre ganger tar det tid.

- *Hva er vi mest fornøyd med?*

Vi er fornøyd med produktListScreen, der fikk vi til å sette inn søke og kategori funksjoner. Vi er også generelt fornøyd med design valgene våre og layouten, vi føler det er et enkelt design med en god sammenheng mellom de ulike skjermene.

Referanseliste:

API:

- FakeStoreAPI. "*FakeStoreAPI*." <https://fakestoreapi.com/> (hentet 14. desember 2023).

Kode:

- Jetpack Compose, Room, ViewModel, Database, DAO. "*PGR208 1 23H Android programmering, Ekstra forelesning, Extra01, Extra02, Extra03, Extra04*."

Emoji:

- EmojiCombos. "*Star Emoji Combos*." <https://emojicombos.com/star> (hentet 11. desember 2023).
- EmojiCopy. "*EmojiCopy*." <https://emojicopy.com/> (hentet 11. desember 2023).
- FSymbols. "*Heart Emoji*." <https://fsymbols.com/heart/> (hentet 12. desember 2023).

Logo og Ikon:

- Icons8. "*Small Business Logo*." <https://icons8.com/icon/53007/small-business> (hentet 12. desember 2023).
- FontAwesome. "*History Icon*." <https://fontawesome.com/v5/icons/history?f=classic&s=solid> (hentet 13. desember 2023).
- Icons8. "*Shopping Icons*." <https://icons8.com/icon/set/shopping/nolan> (hentet 14. desember 2023).

Contrast Checker:

- TPGI. "*Color Contrast Checker*." <https://www.tpgi.com/color-contrast-checker/> (hentet 14. desember 2023).