

# Progetto Wardorbe



## Membri del gruppo:

**Simone Radaelli 902814**

**Raouf Namane 900123**

**Nicholas Azzini 902809**

**Andrea Ranieri 902265**

## Panoramica del progetto

Il progetto ha come scopo la realizzazione di una applicazione mobile Android, attraverso la quale l'utente potrà visionare una serie di prodotti vestiari con annessa descrizione e nome del brand: t-shirt, jeans e sneakers.

Grazie al cuoricino posto in fondo ad ogni prodotto, si ha la possibilità di aggiungerli ai preferiti che potranno visualizzare nella pagina Account. In una sezione dedicata si potrà comporre un outfit a proprio piacimento oppure farlo creare direttamente all'applicazione.

## Tecnologie utilizzate

- **Android Studio:** IDE basato su IntelliJ e usato per sviluppare applicazioni Android in Java.
- **GitHub:** Software utilizzato per gestire la collaborazione tra gli sviluppatori nell'implementazione dell'app. La nostra repository è stata strutturata nel seguente modo:
  - **Branch Main:** il branch principale che contiene le versioni principali dell'app prodotte durante la fase di sviluppo
  - **Branch ohRaouf/SimoRada/nico/andrea:** questi branch sono personali degli sviluppatori, utilizzati per suddividere il lavoro da svolgere a ciascuno dei componenti del gruppo ed al termine del lavoro fare la push sul Branch main.

- **Firebase:** Piattaforma sviluppata da Google che ci permette di gestire i dati dell'applicazione in un database e l'autenticazione dell'utente.
- **RapidAPI:** API esterna che fornisce i dati e le immagini dei capi d'abbigliamento, [Link alla pagina API](#)

Avendo chiamate API limitate e non volendole utilizzare tutte per dei semplici debug, abbiamo implementato una variabile (all'interno delle local properties) chiamata "debug\_mode" che una volta settata a true prende i dati da più file json scaricati localmente. Ovviamente, una volta settata a false, torna a prendere i dati in remoto.

# Design

## Progettazione UI

Per progettare l'interfaccia utente abbiamo utilizzato un tool di progettazione grafica chiamato Figma.

Un tool utilizzato con l'obiettivo di creare delle bozze per le varie interfacce presenti nell'applicazione, in modo da avere una base per poi creare la versione definitiva da implementare su Android Studio.

## Material Design

Lo stile dell'applicazione segue il più possibile le linee guida di Material Design 3, utilizzando e adattando i material components a disposizione, come Buttons e Cards.

Grazie alla palette dei colori di Material Design, abbiamo inoltre implementato due temi: Light Mode e Dark Mode

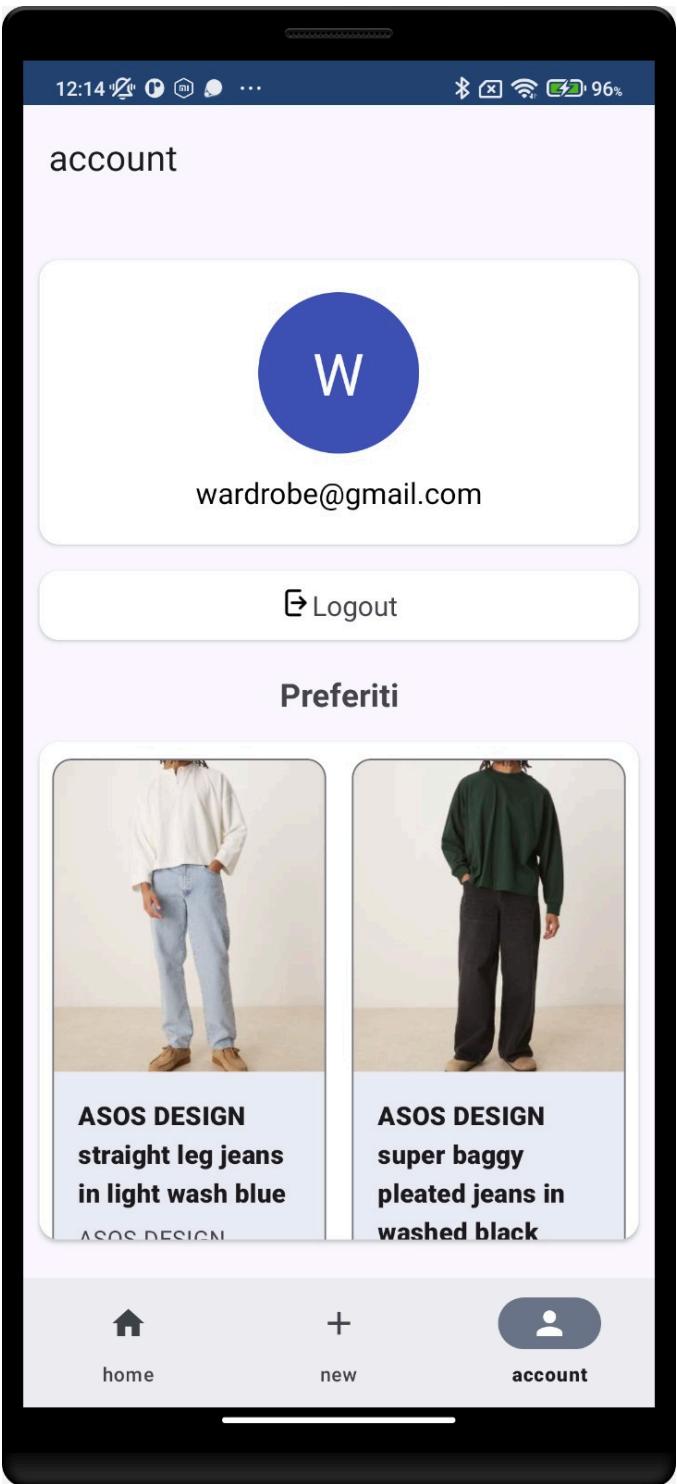


Figura 1: Foto account light mode

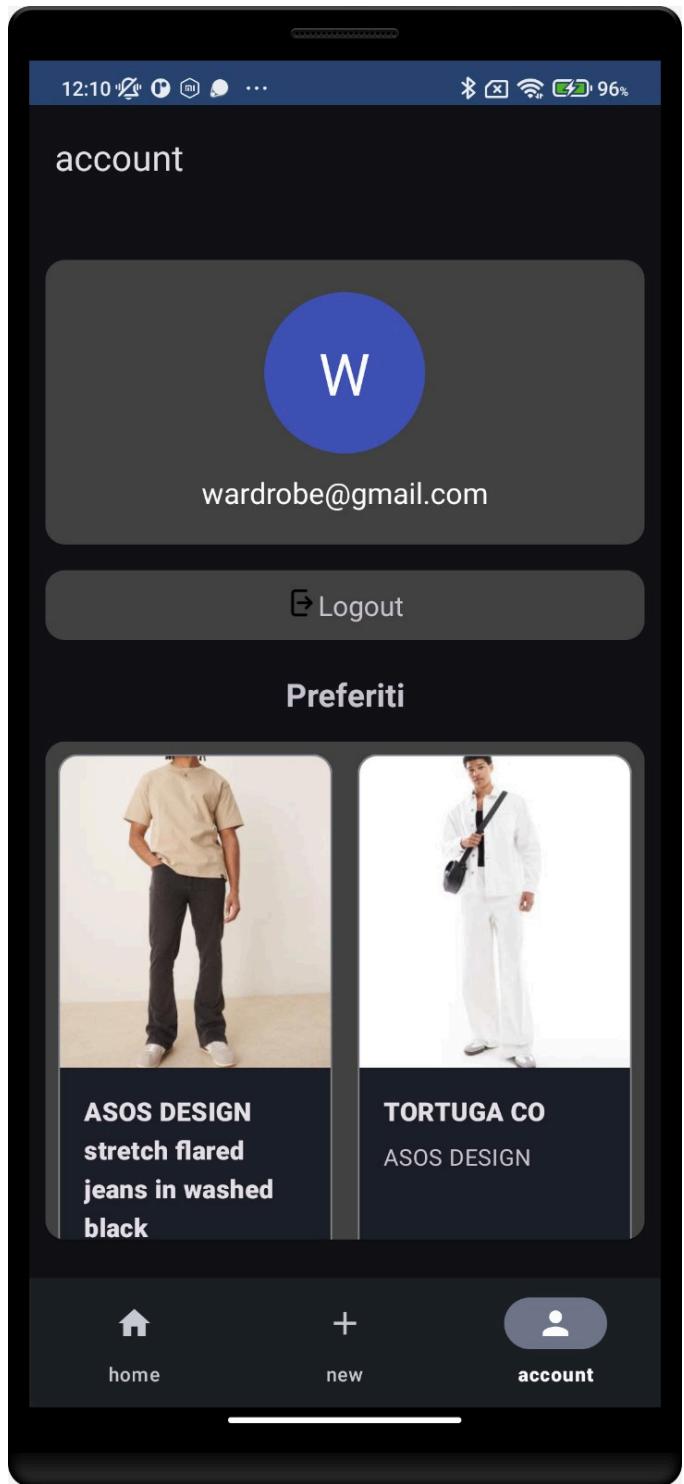
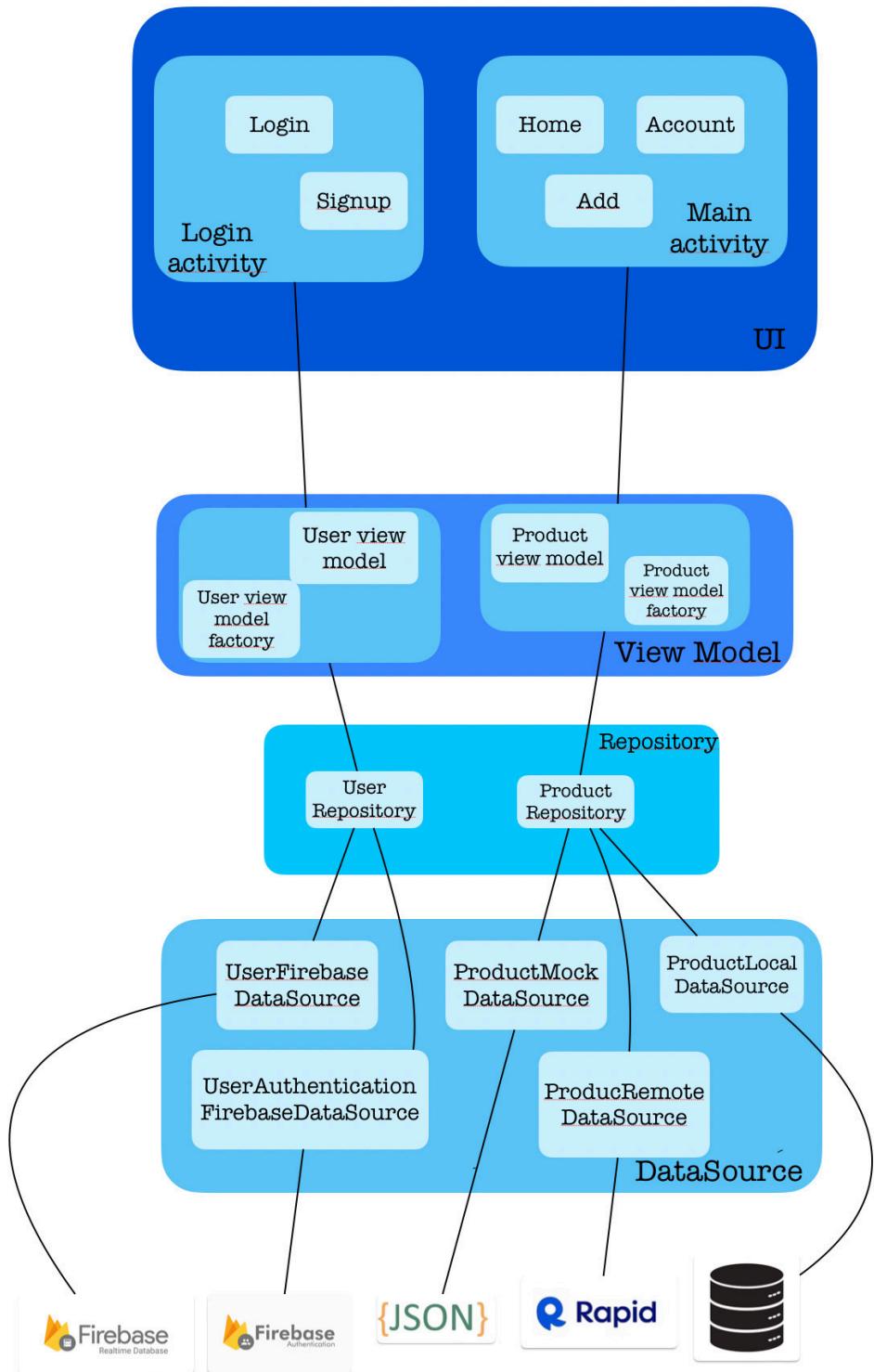


Figura 2: Foto account dark mode

# Architettura



**Figura 3: Foto architettura progetto**

L'applicazione segue l'approccio "Single activity - multiple fragments", in cui ogni schermata è implementata come fragment e gestita attraverso due activity distinte:

- **Welcome Activity:** responsabile dei fragment dedicati all'autenticazione.
- **Home Activity:** gestisce i fragment correlati alla visione dei capi d'abbigliamento e le varie card in cui essi sono, gestione dell'account e della visione dei capi preferiti.

L'architettura adottata si basa sul modello **Model-View-ViewModel (MVVM)** e si compone di diversi elementi principali:

- **UI:** Responsabile dell'interazione con l'utente, della presentazione delle informazioni e dell'aggiornamento dell'interfaccia. Monitora i dati dinamici forniti dal ViewModel attraverso i LiveData.
- **ViewModel:** Gestisce e conserva i dati, assicurando la loro coerenza tramite LiveData. Comunica con l'interfaccia utente per segnalare eventuali variazioni.
  - **UserViewModel:** Si occupa delle operazioni relative agli utenti, tra cui autenticazione tramite Google, gestione di login e logout, e salvataggio dei capi preferiti. Si interfaccia con il repository degli utenti (**IUserRepository**) per recuperare e aggiornare le informazioni.
  - **ProductViewModel:** Fornisce alla UI dati riguardanti liste di prodotti, capi preferiti e outfit generati. Interagisce con **ProductRepository** per ottenere e gestire i dati dei capi e si sincronizza con **IUserRepository** per mantenere aggiornati i preferiti dell'utente autenticato.
- Il **Repository** funge da intermediario tra le sorgenti di dati e il resto dell'applicazione, semplificando l'accesso alle informazioni. Per la gestione dei capi, **ProductRepository** si basa su due fonti distinte:
  - **ProductLocalDataSource:** che opera attraverso **Room** per la memorizzazione e gestione dei dati nel database locale. Notifica il successo o il fallimento delle operazioni tramite callback e mantiene la sincronizzazione dello stato dei capi preferiti memorizzati localmente.
  - **ProductRemoteDataSource:** che esegue chiamate API tramite **RapidAPI** per recuperare dati sui capi. Le richieste asincrone vengono processate e i risultati, positivi o negativi, vengono notificati mediante callback.

## Funzionalità

### Autenticazione:

La schermata di registrazione permette all'utente di accedere al proprio account, inserendo le credenziali e schiacciando il bottone login.

In assenza di un account già registrato, si può andare sulla schermata di registrazione attraverso il bottone del "Sign Up", il quale verificherà l'email grazie alla libreria `CommonsLibraryAccessors`.

La validazione della password, invece, deve solamente rispettare la lunghezza minima di 8 caratteri. Inoltre per permettere un'esperienza migliore all'utente, abbiamo implementato un occhiolino in fondo per mostrare i caratteri inseriti.

Oltre alla registrazione manuale, possiamo entrare direttamente con il nostro account Google tramite il bottone "login with G".



Figura 4: Foto schermata login



Figura 5: Foto schermata Signup

## Home Page

Nella home page è possibile visionare i vari capi suddivisi per categorie con dei caroselli:

- Sneakers
- Jeans
- T-shirt

E' possibile selezionare i capi da inserire all'interno dei preferiti schiacciando l'apposito bottone a forma di cuore che si trova in basso a destra di ogni elemento visualizzato nella schermata.

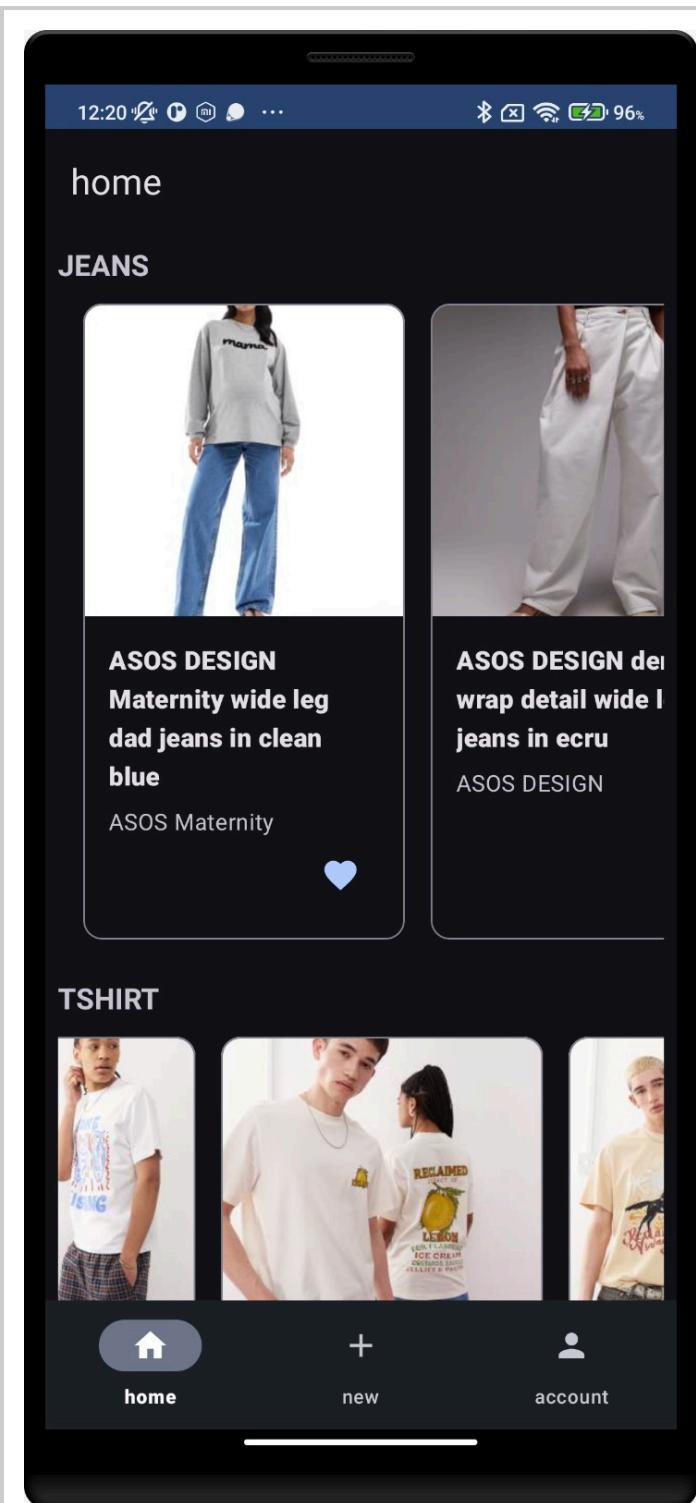


Figura 6: Foto 1 schermata home

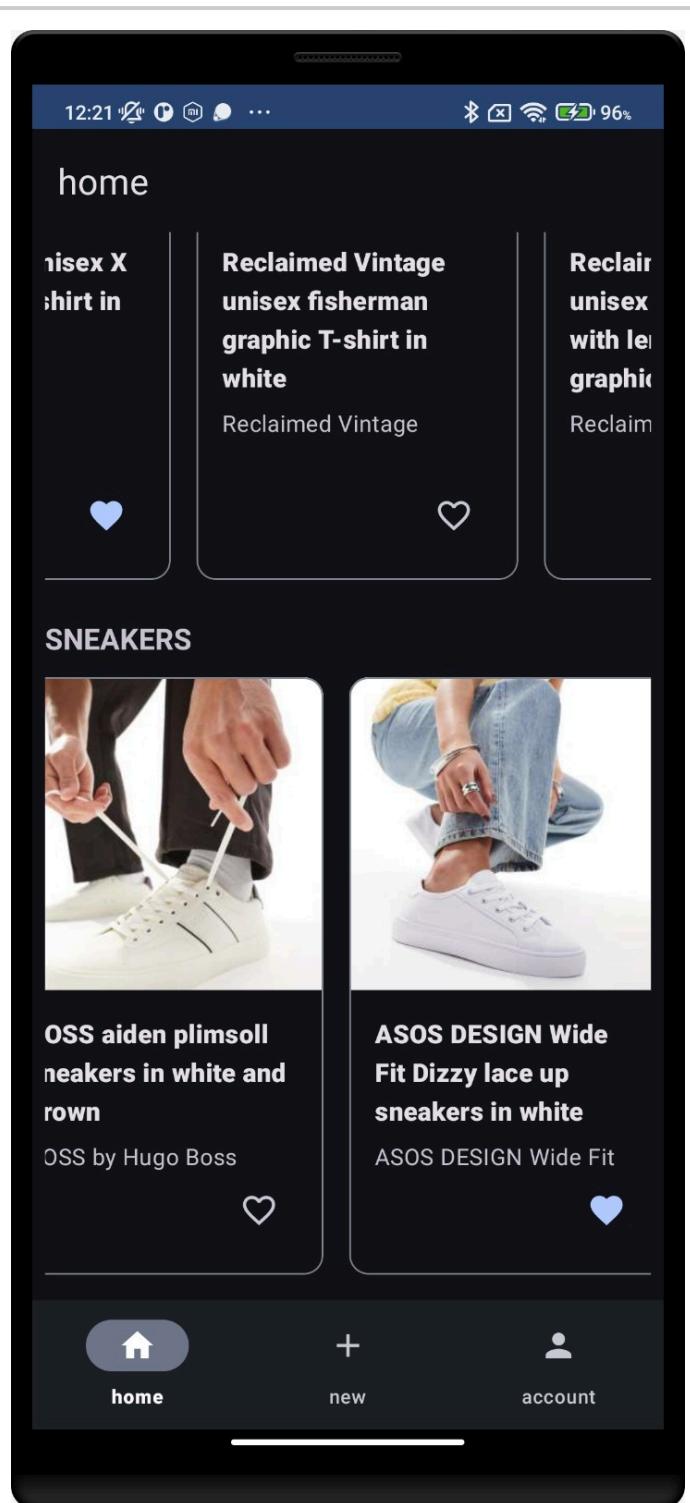
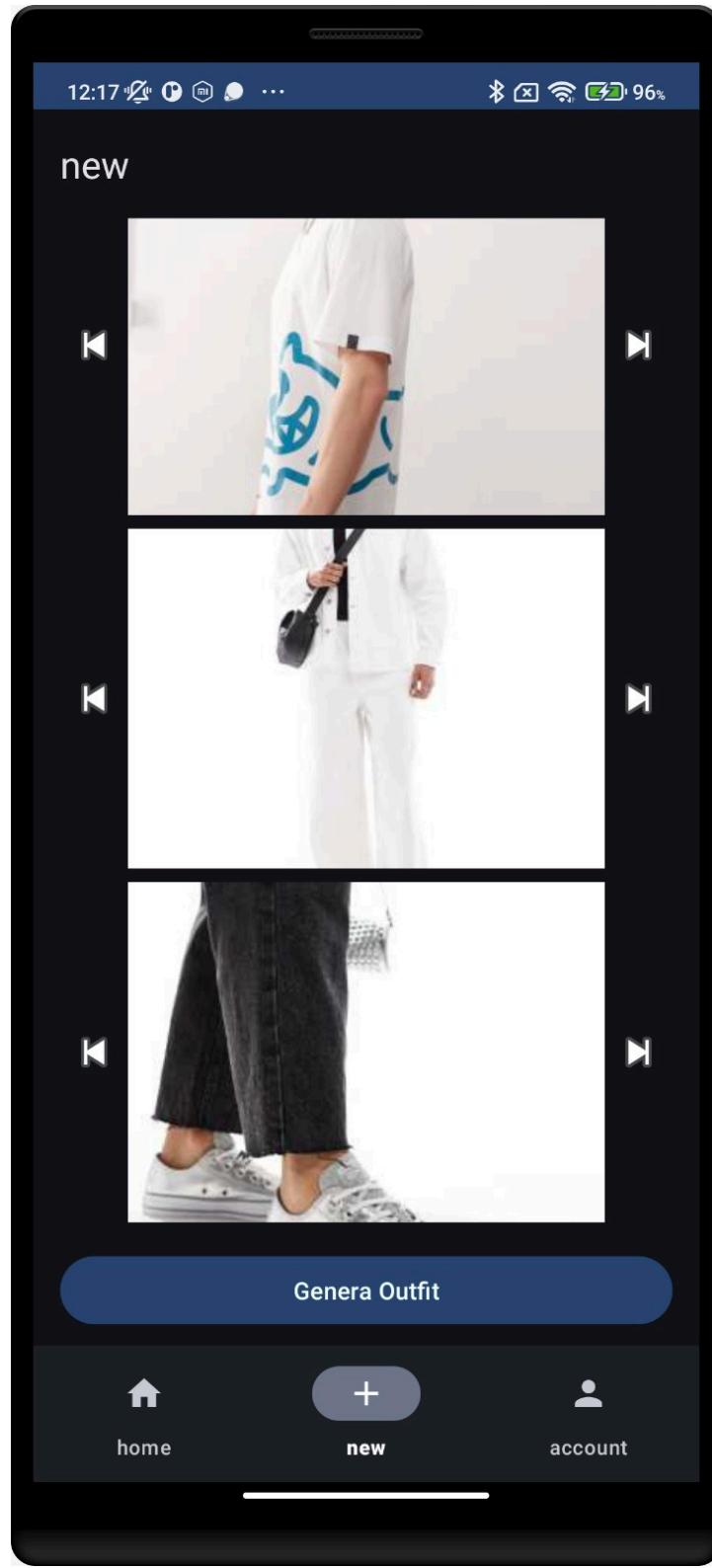


Figura 7: Foto 2 schermata home

## New

Sezione dedicata per la creazione manuale o automatica di un outfit:

Attraverso il bottone **“Genera Outfit”** l'applicazione propone un outfit di base da cui partire, che può essere personalizzato liberamente modificando ogni capo tramite le frecce.

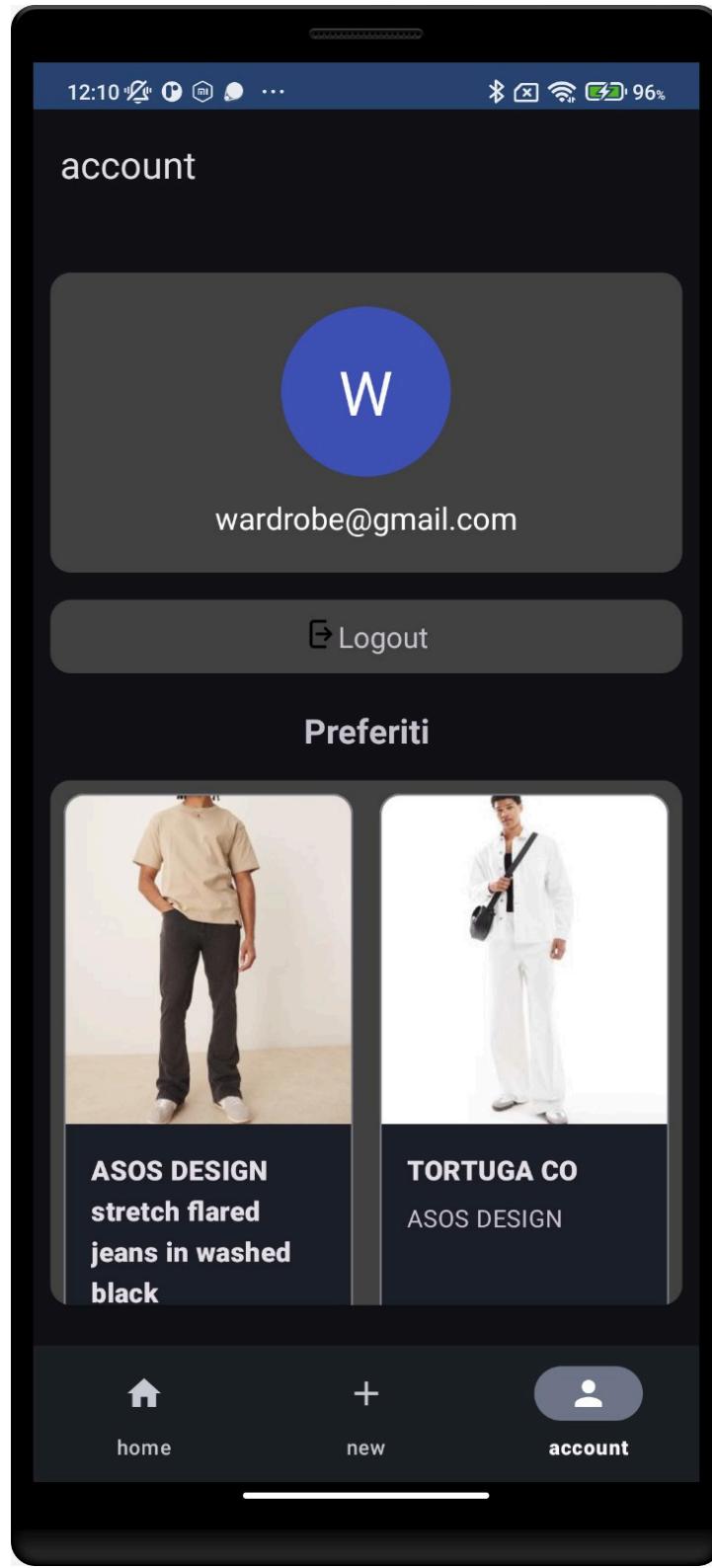


**Figura 8: Foto schermata genera outfit**

## Account

All'interno della sezione **Account** vengono visualizzati:

- L'indirizzo e-mail utilizzato per l'accesso, insieme all'eventuale foto profilo, se presente.
- Il tasto di logout che permette all'utente di disconnettersi dal proprio account.
- Tutti i capi aggiunti ai preferiti, visualizzati in ordine cronologico.



**Figura 9: Foto schermata account**

## Sviluppi futuri

Di seguito varie implementazioni che vorremmo aggiungere per un eventuale release 2.0 in modo da ampliare le funzionalità disponibili, tra le quali:

- Possibilità di aggiungere propri capi all'interno dell'applicazione tramite foto con breve descrizione
- Generazione di outfit abbinati in base al colore o altri parametri
- Ampliamento dei capi disponibili all'interno dell'app
- Implementare il doppio click su un capo per aggiungerlo hai preferiti
- Implementare la possibilità di resettare la password associata all'account creato cliccando sul bottone “**did you forgot your password**” .