

# **Progetto Programmazione Mobile: SoftwareHouse**

## **Indice**

### **1 Presentazione**

*1.1 Presentazione dell'app*

*1.2 Presentazione delle tecnologie*

### **2 Struttura dell'applicazione**

*2.1 Login Activity*

*2.2 Register Activity*

*2.3.1 Home Activity (Admin)*

*2.3.2 InsertNewProduct Activity (Admin)*

*2.3.3 ModifyProduct Activity (Admin)*

*2.3.4 DetailsProduct Activity (Admin)*

*2.4.1 Home Activity (Customer)*

*2.4.2 DetailsProduct Activity (Customer)*

*2.4.3 DatailsProductInWishlist (Customer)*

*2.4.4 Wishlist Activity (Customer)*

### **3 Conclusione**

## **1.1 - Presentazione dell'app**

A completamento del percorso svolto per lo studio della materia “*Programmazione Mobile*”, è stato scelto come progetto la realizzazione di un’applicazione mobile per ambiente “*Android*” tramite la tecnologia “*Android Studio*” con l’utilizzo del linguaggio di programmazione “*Java*” e con l’ausilio di alcuni servizi offerti da Google tramite le tecnologie “*Firebase - Cloud Firestore*”, “*Firebase - Authentication*” e “*Firebase - Storage*”. L’applicazione creata prende il nome di “*SoftwareHouse*” ed il suo scopo è quello di fornire una piattaforma in cui uno o più *Amministratori* possano inserire i propri prodotti digitali che successivamente saranno visualizzabili e gestibili a più utenti che potranno salvarli nella propria *wishlist* e valutarli attraverso un meccanismo di rating.

## **1.2 - Presentazione delle tecnologie**

- Android Studio**

*Android Studio* è una tecnologia rilasciata da Google che ci permette di utilizzare un ambiente di sviluppo integrato adibito per la creazione di applicazioni *Android*.

Esso ci fornisce diversi strumenti utili per la realizzazione di un progetto *Android* permettendoci di ottimizzare sia le prestazioni sia il tempo di sviluppo effettivo poiché tra il ventaglio di utility presenti vi sono tool importantissimi come:

1. Sistema di compilazione e di building automation.
2. Ambiente di Layout editing.
3. Integrazioni e supporto per diversi servizi Google.
4. Gestione di dipendenze e configurazione attraverso i file Gradle e Manifest.

Ma oltre questo, *Android Studio* offre davvero molte altre funzionalità che risultano essere indispensabili nei tempi moderni per la creazione di un progetto *Android*.

- *Firebase - Cloud Firestore***

*Firebase - Cloud Firestore* è un servizio offerto da Google che rientra nella categoria dei servizi *serverless*, permettendo l’accesso ad una suite di

strumenti per archiviare, visualizzare e gestire dati in formato JSON tramite un database NoSQL, sfruttando la scalabilità, l'efficienza e la sicurezza dell'infrastruttura Cloud di Google.

La scelta di utilizzare la più recente tecnologia *Firebase - Cloud Firestore* invece del servizio nativo di Firebase “*Realtime Database*” è legata alla mancanza di necessità di sincronizzazione in tempo reale tra Server e Client, rispetto ai più moderni strumenti forniti da Firestore che permettono di utilizzare un modello di dati più intuitivo ed un set di query più complete e veloci fornendo inoltre una scalabilità maggiore rispetto ai database in tempo reale.

Tramite *Firebase* ed i suoi strumenti è stato possibile gestire i dati raccolti ed elaborati all'interno dell'applicazione *SoftwareHouse*, attraverso operazioni di scrittura, lettura e modifica nel DB offerto da Firestore stesso.

Le operazioni eseguite vengono autorizzate tramite un meccanismo di *Regole (Rules)* impostate opportunamente all'interno della dashboard di *Firebase*.

```
service cloud.firestore {
  match /databases/{database}/documents {
    match /Users/{userId} {
      allow read: if request.auth != null;
      allow write: if request.auth != null;
    }

    match /Products/{product} {
      allow read: if request.auth != null;
      allow write: if request.auth != null;
    }
  }
}
```

Nello specifico le regole sopra descritte autorizzano le varie richieste di lettura e scrittura (divise in inserimento e modifica) ricevute dal Database solo se il token d'autorizzazione fornito risulta essere valido, questo è possibile grazie all'utilizzo di un secondo servizio offerto da *Firebase* che prende il nome di “*Firebase - Authentication*”.

- **Firebase - Authentication**

*Firebase - Authentication* è un servizio d'autenticazione offerto da Google che permette di gestire l'identità dei vari Client che effettuano richieste di lettura e scrittura al Database.

Per l'acquisizione di un token valido e la sua associazione è necessario sottoscrivere il proprio Account al servizio di Firebase - Authentication, questo meccanismo viene regolamentato da un servizio email automatizzato che impone all'utente di confermare la propria identità attraverso un'email inviata all'indirizzo di posta elettronica specificato in fase di registrazione. Questo ci permetterà di autorizzare il login e le diverse operazioni consentite sul Database ai soli account già verificati.

- **Firebase - Storage**

*Firebase - Storage* è un servizio offerto da Google per l'archiviazione dei documenti multimediali generati dagli utenti.

Per l'utilizzo di Firebase Storage è stato creato un solo sotto ramo di possibili tipologie di risorse in quanto all'interno dell'applicazione SoftwareHouse sarà possibile caricare solo immagini a descrizione dei prodotti inseriti.

gs://softwarehouse-d8b9d.appspot.com

Nome

 images/

## 2 - Struttura dell'applicazione

La struttura di *SoftwareHouse* è divisa in due macro insiemi di risorse (Activity, Menu, Layout, Drawable, ecc...) che identificano un differente funzionamento dell'applicazione in base all'utilizzo di quest'ultima da parte di un *Amministratore* o di un *Utente Semplice*.

La distinzione dell'entità che desidera utilizzare l'applicazione avverrà in fase di login, nello specifico tramite la gestione del token univoco fornito da Firebase - Authentication.

Per comprendere la struttura analizzeremo le Activity principali che compongono l'applicazione *SoftwareHouse*.

### 2.1 - Login Activity

Elementi grafici del layout:

- CardView (logo App)
- TextView
- EditText
- Button
- ConstraintLayout

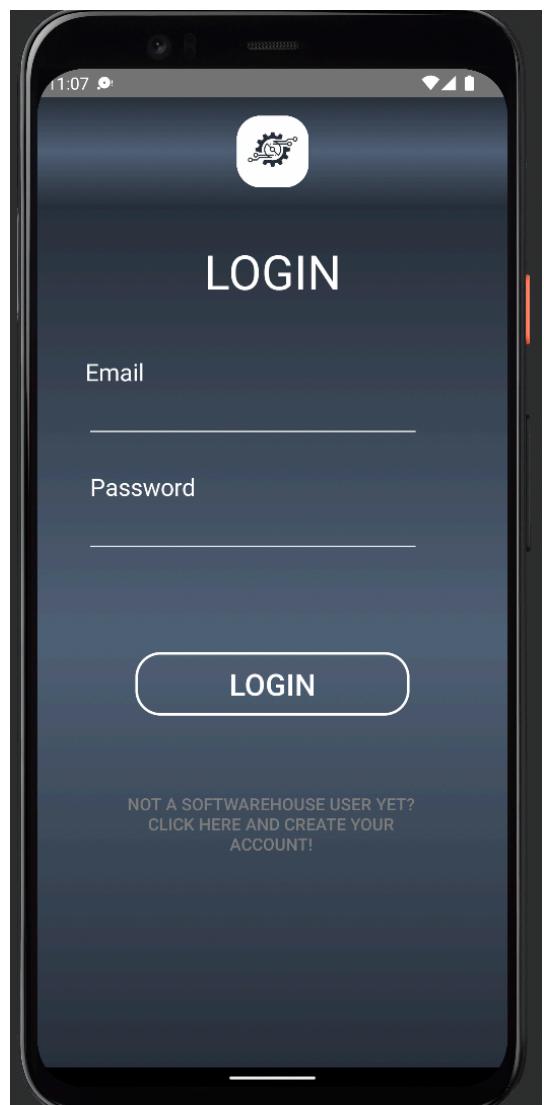
L'activity per la gestione del login è la prima activity visualizzata all'apertura dell'applicazione.

Tramite essa possiamo inserire le credenziali del nostro account già precedentemente creato e verificato tramite email, per effettuare il login.

Questo avverrà tramite l'utilizzo di un oggetto di tipo *FirebaseAuth*, che ci permetterà di conservare ed utilizzare l'istanza del servizio Firebase - Authentication.

Verrà utilizzato anche un oggetto *FirebaseFirestore* che ci permetterà di recuperare i dati dell'utente qualora le credenziali inserite fossero valide

Nel caso *in cui* non possedessimo ancora un account valido, potremo cliccare sulla scritta in fondo all'activity per procedere con la fase di registrazione.

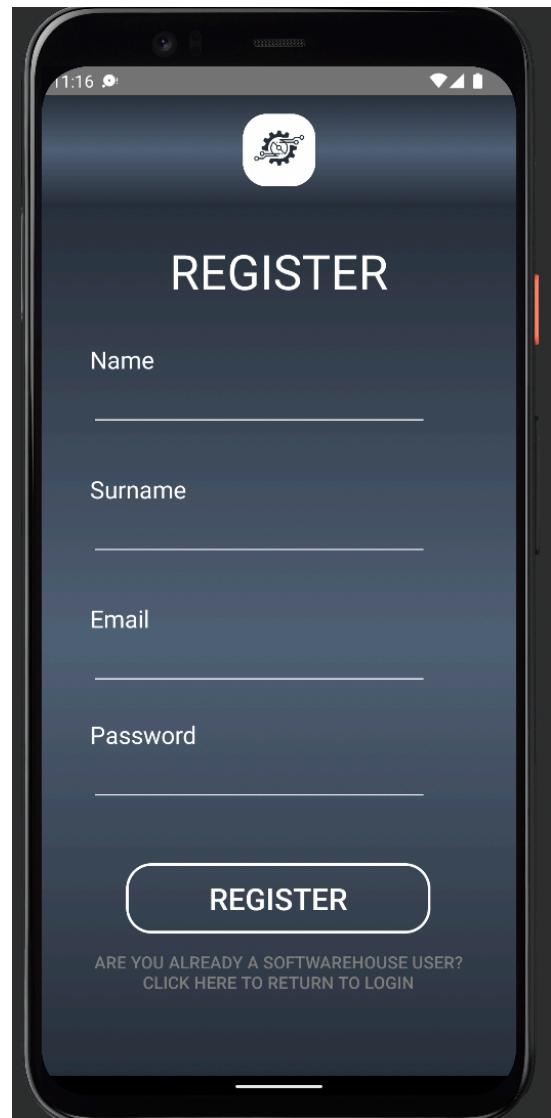


## 2.2 - Register Activity

Elementi grafici del layout:

- CardView (logo App)
- TextView
- EditText
- Button
- ConstraintLayout

L'activity per la gestione della registrazione permetterà all'utente di inserire i propri dati personali, successivamente tramite le istanze ottenute dei servizi *FirebaseAuth* e *FirebaseFirestore* controlleremo l'univocità dell'indirizzo inserito ed in caso di indirizzo valido procederemo con l'invio dell'email per la validazione dell'account e all'inserimento dei dati acquisiti dall'utente nel nostro database.



### 2.3.1 - Home Activity (Admin)

Elementi grafici del layout:

- CardView (logo App)
- TextView
- EditText
- Button
- ImageButton
- ConstraintLayout
- RecyclerView (Lista dei prodotti)
- RecyclerViewAdapter (Lista dei prodotti)
- DrawerLayout (Menu Laterale)
- ScrollView (Vertical)
- NavigationView (Menu Laterale)
- SearchBar

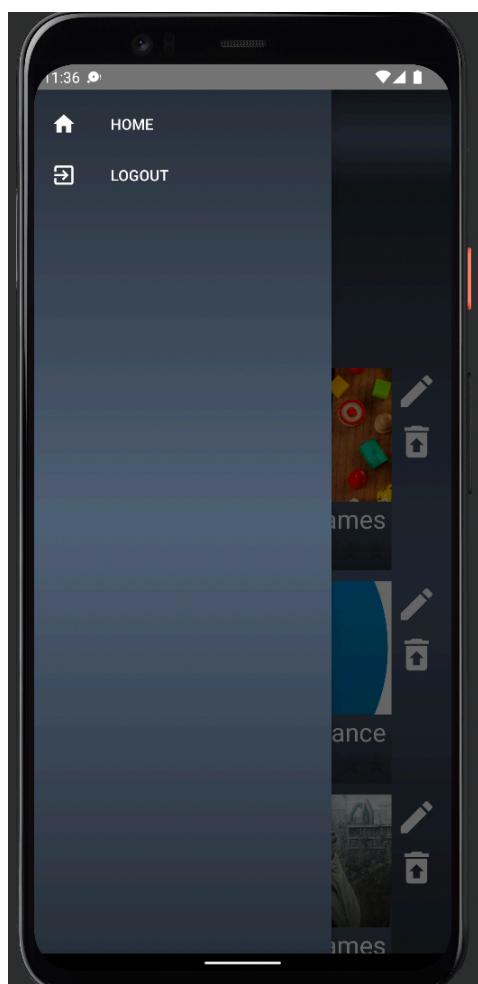
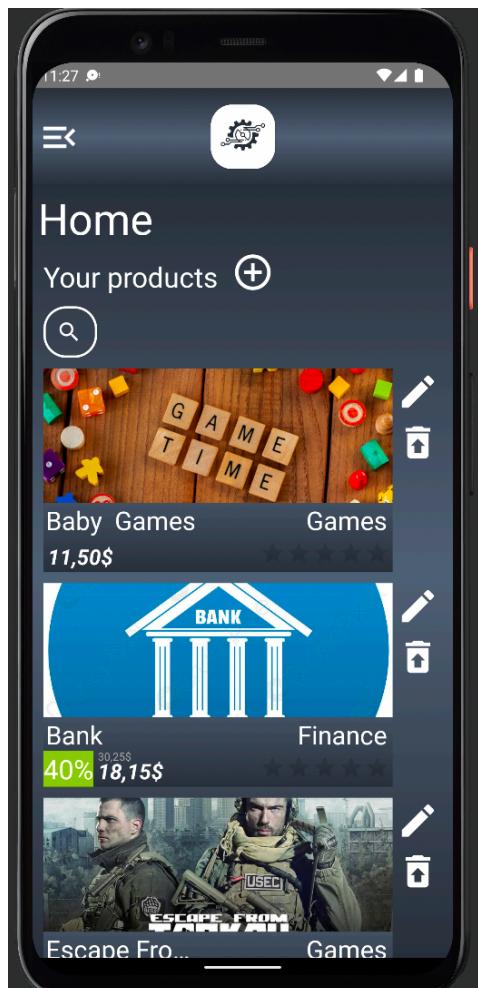
L'activity per la visualizzazione della Homepage dell'utente *Amministratore* presenta un layout con diversi elementi che permettono la visualizzazione e la gestione dei prodotti digitali.

L'amministratore può creare, modificare e rimuovere i prodotti presenti nell'elenco ordinato in ordine alfabetico oppure visualizzare i dettagli di uno di essi cliccando sulla sua immagine.

E' possibile la ricerca di un prodotto con un *Titolo* specifico attraverso una barra di ricerca che tramite posta prima della lista.

Tramite una gestione laterale o al click del pulsante alla sinistra della barra principale sarà possibile aprire un menu laterale che ci permetterà di effettuare il Logout.

*N.B:* Non è stato permesso all'utente di modificare o interagire con la barra di valutazione dei prodotti così da non interferire con la media aritmetica generata dai soli utenti.



## 2.3.2 - InsertNewProduct Activity (Admin)

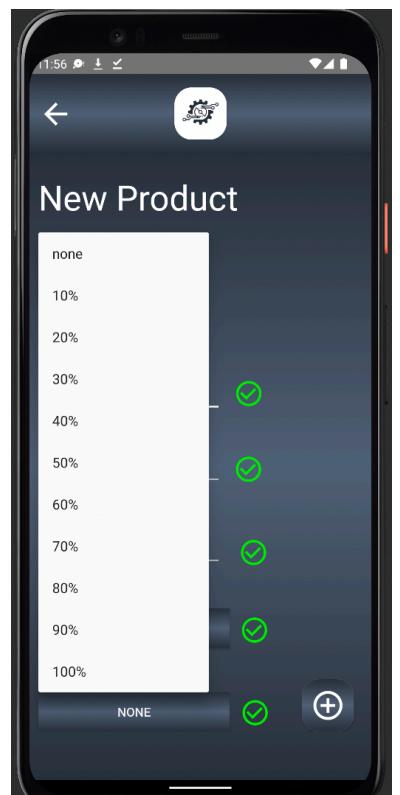
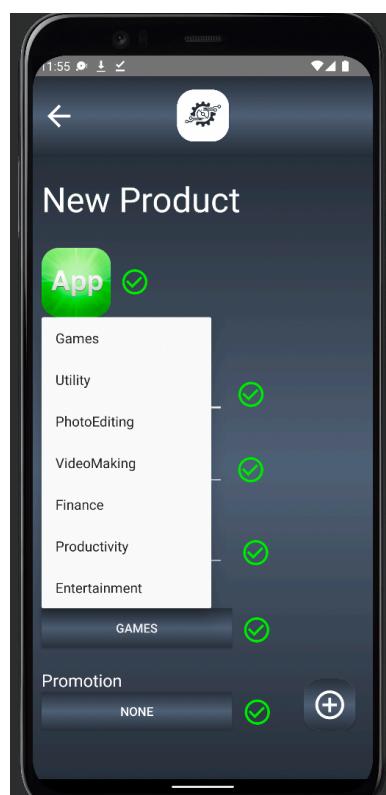
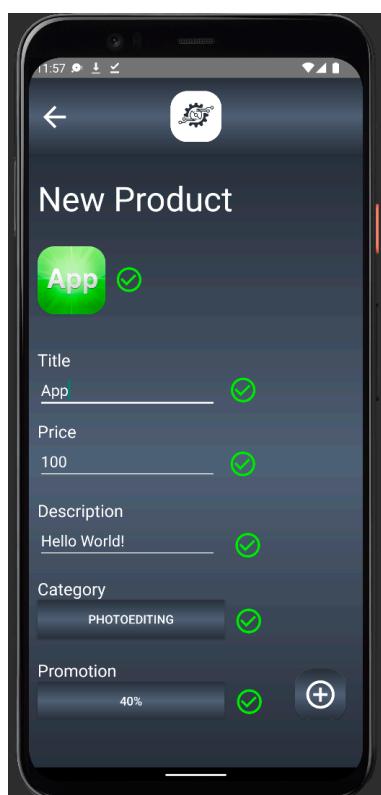
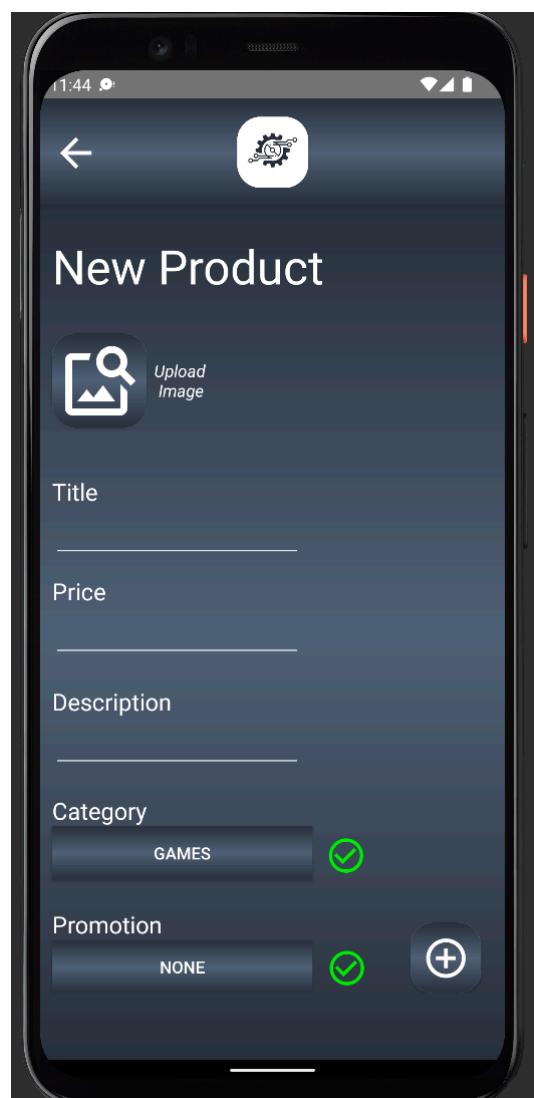
Elementi grafici del layout:

- CardView (logo App)
- TextView
- EditText
- Button
- ImageButton
- ConstraintLayout
- Menu (Category, Promotion)
- ScrollView

L'activity per la gestione dell'inserimento di un nuovo prodotto da parte dell'*amministratore* permette l'inserimento dei dati tipici per la sua descrizione e l'aggiunta di una Categoria (già pre impostata) e di una possibile promozione da applicare al nuovo prodotto, queste ultime due proprietà possono essere aggiunte da due menu contestuali cliccando sui bottoni Category e Promotion.

Per l'inserimento del nuovo prodotto sarà necessario inserire anche un'immagine selezionabili dalla galleria dell'*amministratore*.

Di seguito vengono visualizzate le view dei menu contestuali e dei campi riempiti correttamente:



*N.B:* Non vengono richieste autorizzazione per l'accesso alla fotocamera in fase d'inserimento dell'immagine poiché è già stato concesso in quanto account Amministratore.

### 2.3.3 - ModifyProduct Activity (Admin)

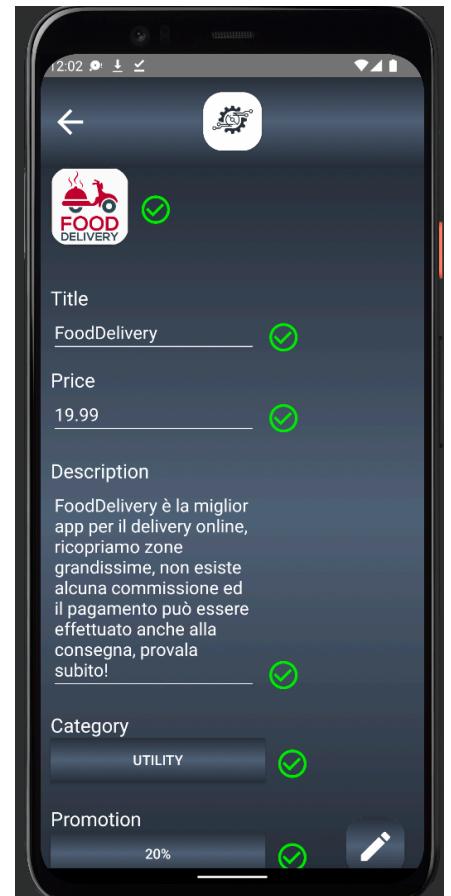
Elementi grafici del layout:

- CardView (logo App)
- TextView
- EditText
- Button
- ImageButton
- ConstraintLayout
- ScrollView
- Menu (Category, Promotion)

Simile all'activity per l'inserimento di un nuovo prodotto, l'activity per la modifica ci permette di visualizzare i dati già presenti del prodotto selezionato e di modificarne opportunamente qualsiasi campo.

Gli elementi grafici ed il loro funzionamento sono pressoché identici all'activity per l'inserimento dunque non verranno trattati in questa sezione.

L'aggiornamento del prodotto avverrà al click del bottone per la modifica in fondo a destra.



### 2.3.4 - DetailsProduct Activity (Admin)

Elementi grafici del layout:

- CardView (logo App)
- TextView
- EditText
- ConstraintLayout
- ScrollView
- RatingBar

L'activity per la visualizzazione dei dettagli mostra in un formato migliore e più semplice i dati di un prodotto visualizzando, a differenza della home iniziale, anche la descrizione del prodotto.



Essa non permette alcuna interazione se non quella del ritorno alla home tramite il bottone posto alla sinistra della barra principale.

### 2.4.1 - Home Activity (Customer)

Elementi grafici del layout:

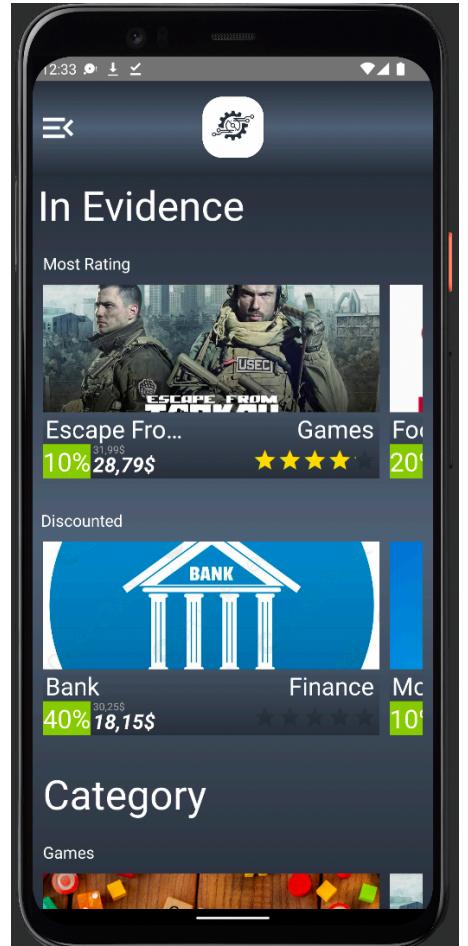
- CardView (logo App)
- TextView
- Button
- ImageButton
- ConstraintLayout
- RecyclerView (Lista dei prodotti)
- RecyclerViewAdapter (Lista dei prodotti)
- DrawerLayout (Menu Laterale)
- ScrollView (Vertical)
- NavigationView (Menu Laterale)

L'activity per la visualizzazione della Homepage dell'utente *Customer* presenta un layout ricco di liste di prodotti divise per due macro sezioni:

#### 1. In Evidence

1.1 Most Rating: sono presenti tutti i prodotti con una valutazione maggiore o uguale a 4 stelle.

1.2 Discounted: sono presenti tutti i prodotti in saldo.



#### 2. Category

- 2.1 Categoria Games
- 2.2 Categoria Utility
- 2.3 Categoria PhotoEditing
- 2.4 Categoria VideoMaking
- 2.5 Categoria Finance
- 2.6 Categoria Productivity
- 2.7 Categoria Entertainment



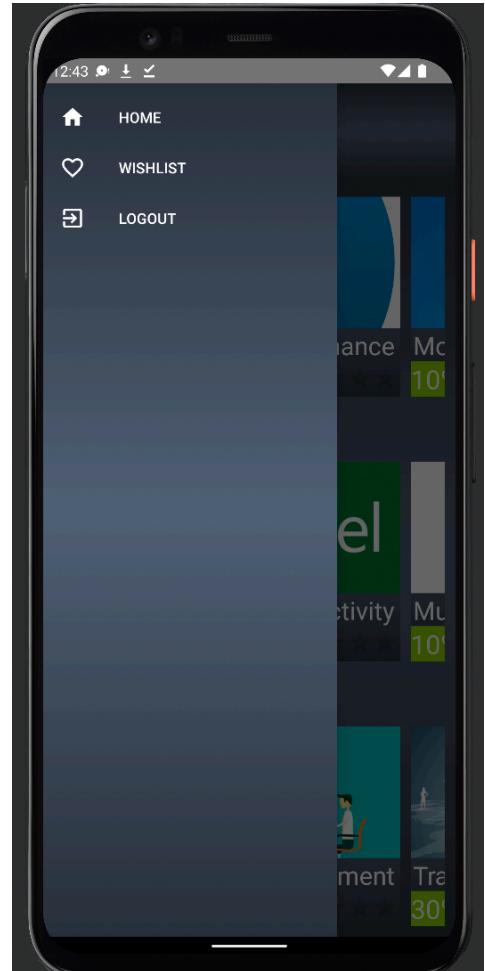
Le liste sono una combinazione di RecyclerView orizzontali inserite all'interno di una ScrollView verticale che permette lo scorrimento di tutte le possibili categorie viste che quest'ultime vengono create dinamicamente solo se nel Database è

presente un prodotto appartenente ad una determinata categoria.

I Prodotti appaiono tutti in card dinamiche in cui se presente uno sconto promozionale il prezzo verrà visualizzato assieme allo sconto applicato ed al prezzo precedente (in grigio e più piccolo), se invece non vi è alcuno sconto il prodotto verrà visualizzato senza queste informazioni aggiuntive.

L'utente *Customer* potrà selezionare qualsiasi prodotto tramite la sua immagine navigando alla pagina dei dettagli del prodotto in cui potrà aggiungere quest'ultimo alla sua *wishlist*.

Tramite una gestione laterale oppure al click del bottone posto alla sinistra della barra principale si aprirà un menu laterale che ci permetterà all'utente di raggiungere la propria wishlist oppure effettuare il Logout.



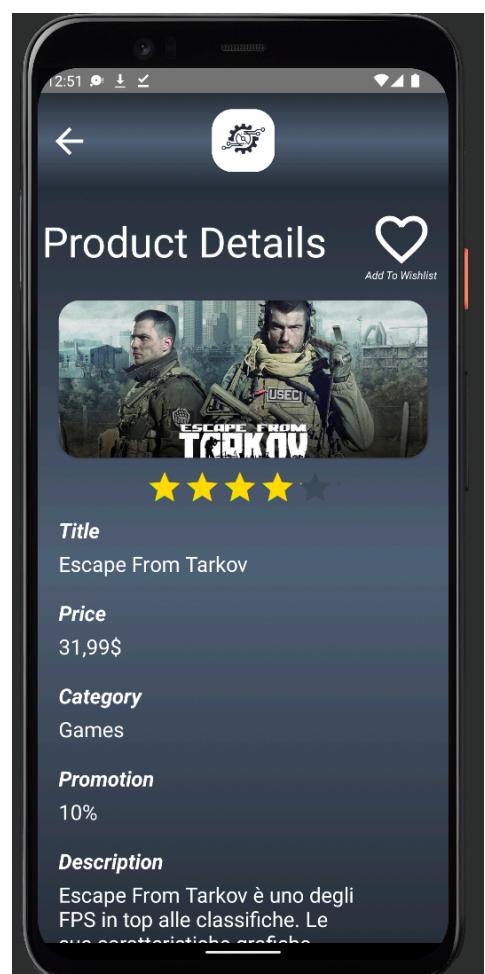
## 2.4.2 - DetailsProduct Activity (Customer)

Elementi grafici del layout:

- CardView (logo App)
- TextView
- ImageButton
- ConstraintLayout
- ScrollView
- RatingBar

L'activity per la visualizzazione dei dettagli per l'utente *customer* risulta uguale a quella dell'utente *amministratore* con la differenza di poter aggiungere il prodotto visualizzato attualmente nella propria *wishlist*.

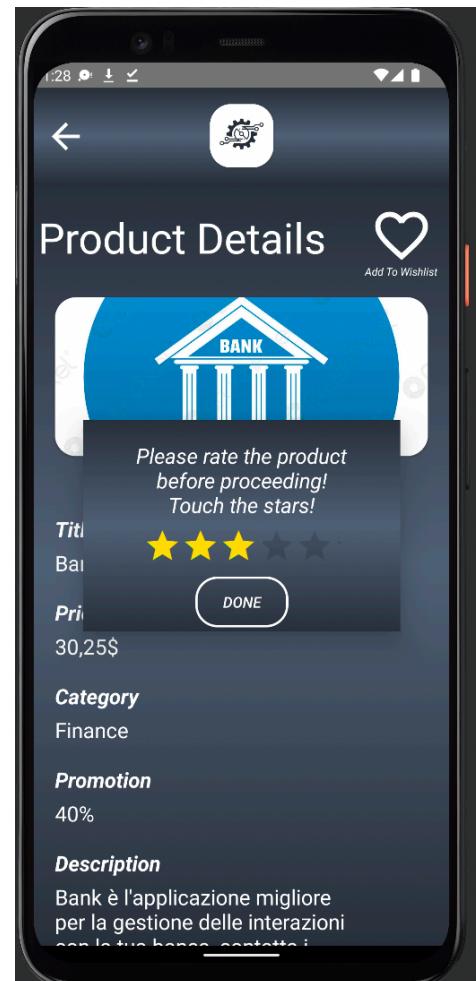
Per farlo l'utente potrà cliccare nell'icona in alto a destra a forma di cuore, una volta fatto verrà visualizzata una schermata temporanea per



l'inserimento del rating obbligatorio per aggiungere l'elemento alla propria wishlist.

Il valore del rating di un prodotto va da 0 a 5, ed è identificato dal numero di stelle indicate prima del click sul pulsante “Done” che concluderà l'operazione.

N.B: La scelta di far inserire all'utente il voto in fase di inserimento del prodotto nella propria wishlist è resa necessaria al fine di evitare che un utente possa non votare un prodotto presente nella sua wishlist, questo meccanismo sarebbe stato inserito correttamente in un possibile carrello dopo aver effettuato l'acquisto e l'applicazione l'avesse richiesto.

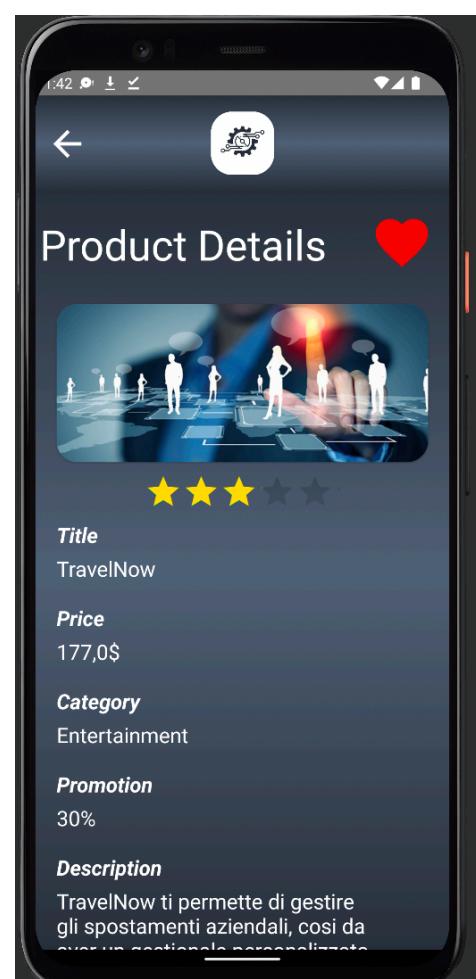


#### 2.4.3 - DetailsProductInWishList (Customer)

Elementi grafici del layout:

- CardView (logo App)
- TextView
- ImageButton
- ConstraintLayout
- ScrollView
- RatingBar

Se l'utente visualizza un prodotto presente nella propria wishlist allora questo verrà segnalato dal cuore rosso in alto a destra nell'activity e non sarà più possibile aggiungerlo oppure votarlo nuovamente.



## 2.4.4 - Wishlist Activity (Customer)

Elementi grafici del layout:

- CardView (logo App)
- TextView
- Button
- ImageButton
- ConstraintLayout
- RecyclerView (Lista dei prodotti)
- RecyclerViewAdapter (Lista dei prodotti)
- DrawerLayout (Menu Laterale)
- ScrollView (Vertical)
- NavigationView (Menu Laterale)

L'activity per la visualizzazione della wishlist è formata da una RecyclerView verticale che permette di visualizzare tutti i prodotti aggiunti in precedenza nella wishlist da parte dell'utente.

Selezionando un prodotto sarà possibile visualizzare i suoi dettagli come visto in precedenza.

L'utente potrà accedere ad un menu laterale tramite una gestione laterale oppure al click del pulsante alla sinistra della barra principale, questo permetterà all'utente di ritornare nella home principale o di effettuare il logout.



### **3 - Conclusione**

L'applicazione *SoftwareHouse* conta in totale le seguenti risorse:

*Activities*: 11.

*Layouts*: 13.

*Menus*: 4.

*Drawables*: 34.

*Dependencies*: 16.

Nelle future modifiche si potrebbe implementare un meccanismo d'acquisto al fine di ricevere un codice univoco per poter riscattare il prodotto comprato ed effettuare il download direttamente dalla Dashboard dell'azienda che lo fornisce o tramite lo store mobile.

*Docente: Massimo Orazio Spata.*

*Studente: Dinatale Alessio.*

*Matricola: X81000589.*