

Sommario

Implementazione	2
Implementazione del software	2
Istruzioni per il deploy del software	2
Configurazione Android Studio per il collegamento con il database	2
Organizzazione DB NoSQL Firebase	3
Esecuzione software BuyGames.....	6
Home.....	6
Lista giochi	7
Dettaglio gioco	8
Inserimento	9
Login.....	9

Implementazione

Implementazione del software

L'applicazione BuyGames è stata sviluppata utilizzando l'ambiente di sviluppo integrato (IDE) *Android Studio* per lo sviluppo per la piattaforma Android.

I linguaggi utilizzati sono:

- *Kotlin* per la parte di back-end
- *XML* per la parte di front-end

In particolare, Kotlin risulta essere pienamente operativo con l'ambiente Java, superandone le limitazioni e le criticità. Inoltre, garantisce il funzionamento delle applicazioni in ogni ambiente che accetti la JVM, ivi compreso Android.

XML, invece, serve a definire le strutture della *user interface* all'interno delle applicazioni fornendo un modo semplice di costruire le View con una serie di elementi innestati.

Istruzioni per il deploy del software

Per effettuare il *deploy* dell'applicazione BuyGames occorre unicamente uno *smartphone* Android sul quale installare l'*apk* ed una connessione ad Internet per interagire con il *database*.

Configurazione Android Studio per il collegamento con il database

Come database esterno di supporto all'applicazione si è scelto di utilizzare il DB NoSQL fornito da Google: *Firebase*. In particolare, Google fornisce una serie di librerie client utilizzate quando si effettuano le richieste tramite API, i servizi offerti da Firebase sono REST e fanno uso del protocollo HTTPS.

Il collegamento tra l'applicazione e il DB aggiungendo al file *build.gradle*

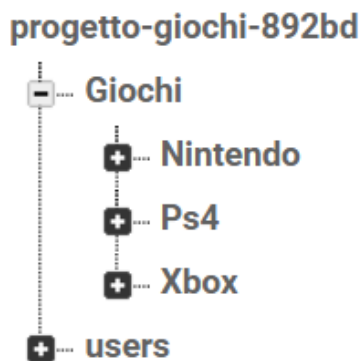
```
implementation 'com.google.firebase:firebase-auth:16.0.5'
implementation 'com.google.firebase:firebase-database:16.0.4'
implementation 'com.google.firebase:firebase-storage:16.0.4'

// FirebaseUI Storage only
implementation 'com.firebaseui:firebase-ui-storage:4.3.1'
implementation 'com.github.bumptech.glide:glide:4.9.0'
kapt 'com.github.bumptech.glide:compiler:4.9.0'
}
```

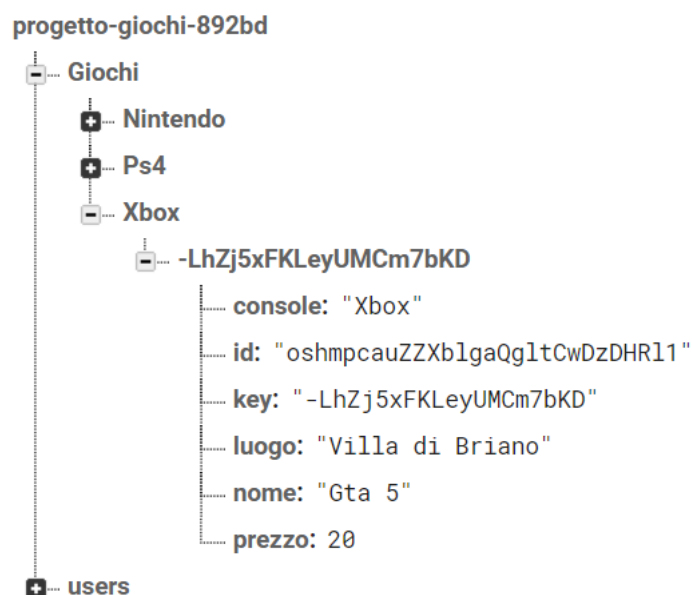
Organizzazione DB NoSQL Firebase

Il database è diviso a sua volta in due database:

- Realtime Database, il quale contiene informazioni riguardanti gli attributi dei giochi e degli utenti. Il database è organizzato ad albero e si divide in Giochi ed Utenti ed è possibile notare che Giochi è diviso a sua volta in base alla piattaforma:



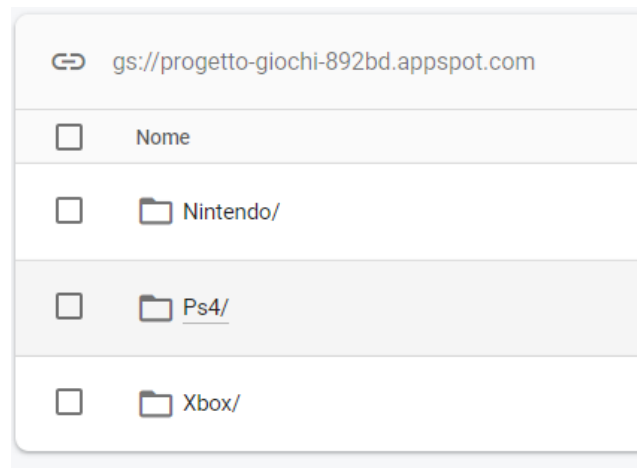
All'interno di ogni piattaforma i giochi sono salvati per *key* e, per ogni gioco, sono salvati i valori di tutti gli attributi:



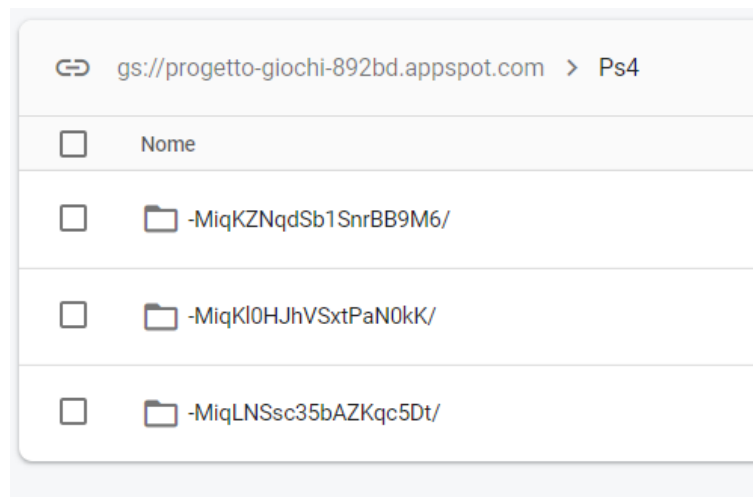
Per quanto riguarda gli utenti, invece, sono memorizzati per Id e contengono sia i dati utente che la lista di giochi memorizzati per *key*, come visto nel ramo *Giochi*:



- Storage Database, il quale contiene le immagini relative ai giochi; è organizzato sempre in 3 cartelle, una per ogni piattaforma:



All'interno di ogni cartella sono presenti i vari giochi salvati per *key*:



Ed infine, per ogni gioco sono presenti le immagini salvate:

gs://progetto-giochi-892bd.appspot.com > Ps4 > -MiqKZNqdSb1...

[Carica file](#)

<input type="checkbox"/>	Nome	Dimensioni	Tipo	Ultima modifica
<input type="checkbox"/>	picture0	50.62 KB	image/jpeg	5 set 2021
<input type="checkbox"/>	picture1	50.08 KB	image/jpeg	5 set 2021
<input type="checkbox"/>	picture2	49.78 KB	image/jpeg	5 set 2021

picture0

Nome
[picture0](#)

Dimensioni
51.838 byte

Tipo
image/jpeg

Data di creazione
5 set 2021, 16:21:53

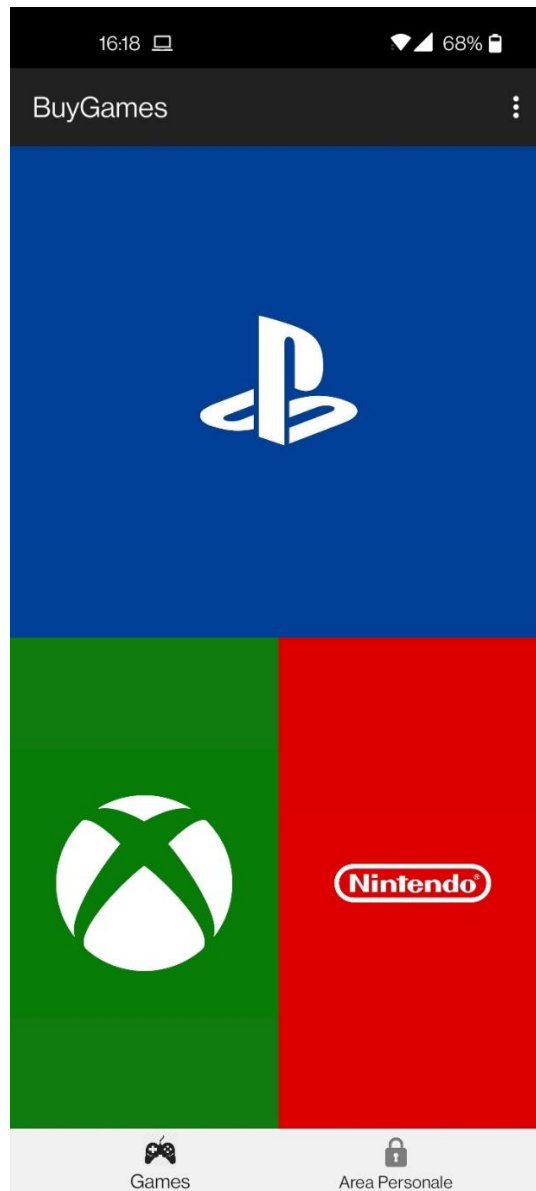
Data di aggiornamento
5 set 2021, 16:21:53

Esecuzione software BuyGames

Il seguente paragrafo riporta schermate del *software* in esecuzione, ovvero ha lo scopo di mostrare come sono state implementate le varie *view*.

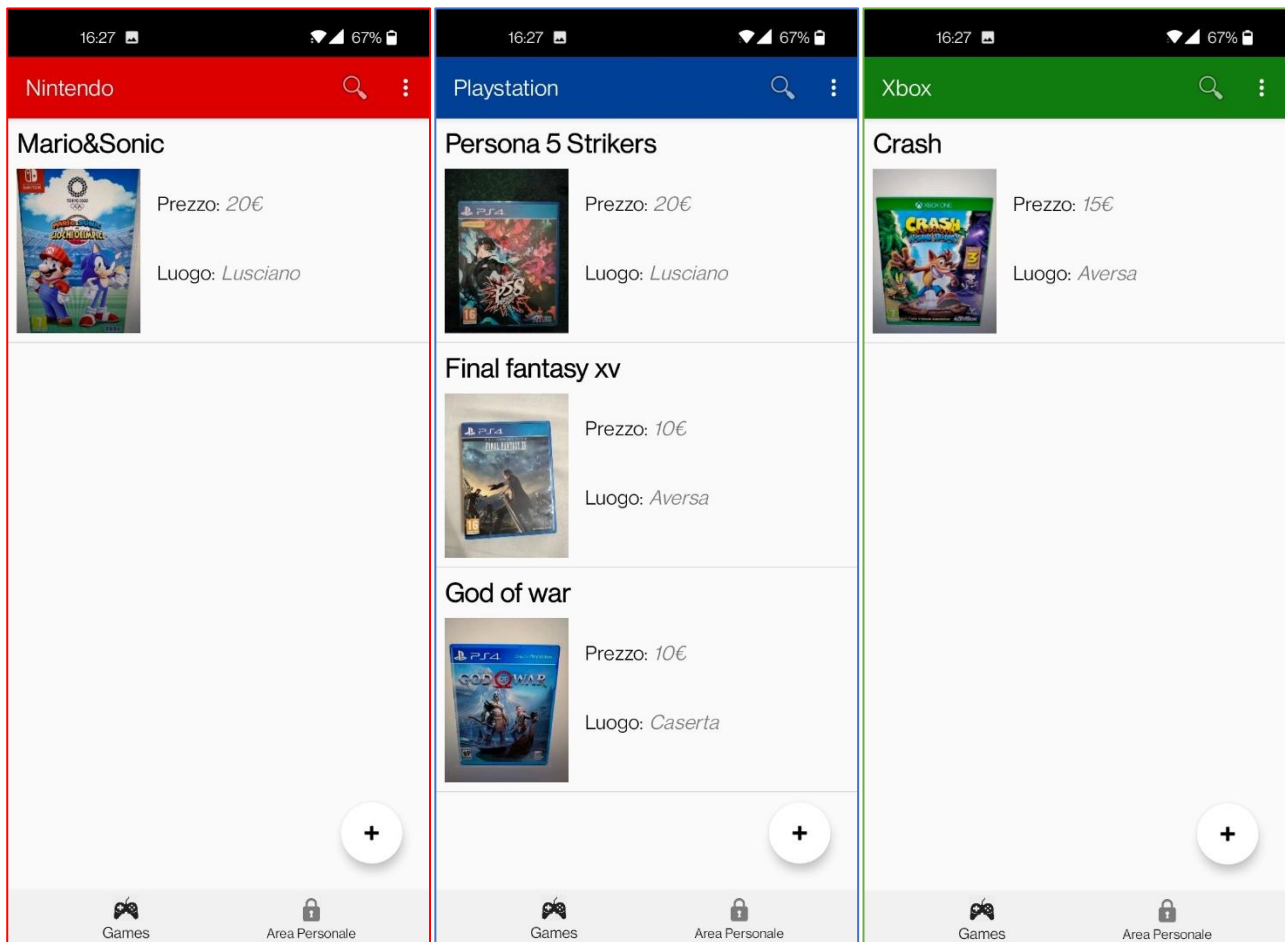
Home

In questa *view* è possibile selezionare la piattaforma per poi ottenere la lista dei giochi in vendita.



Lista giochi

Una volta selezionata la piattaforma si otterrà una delle seguenti liste di giochi:



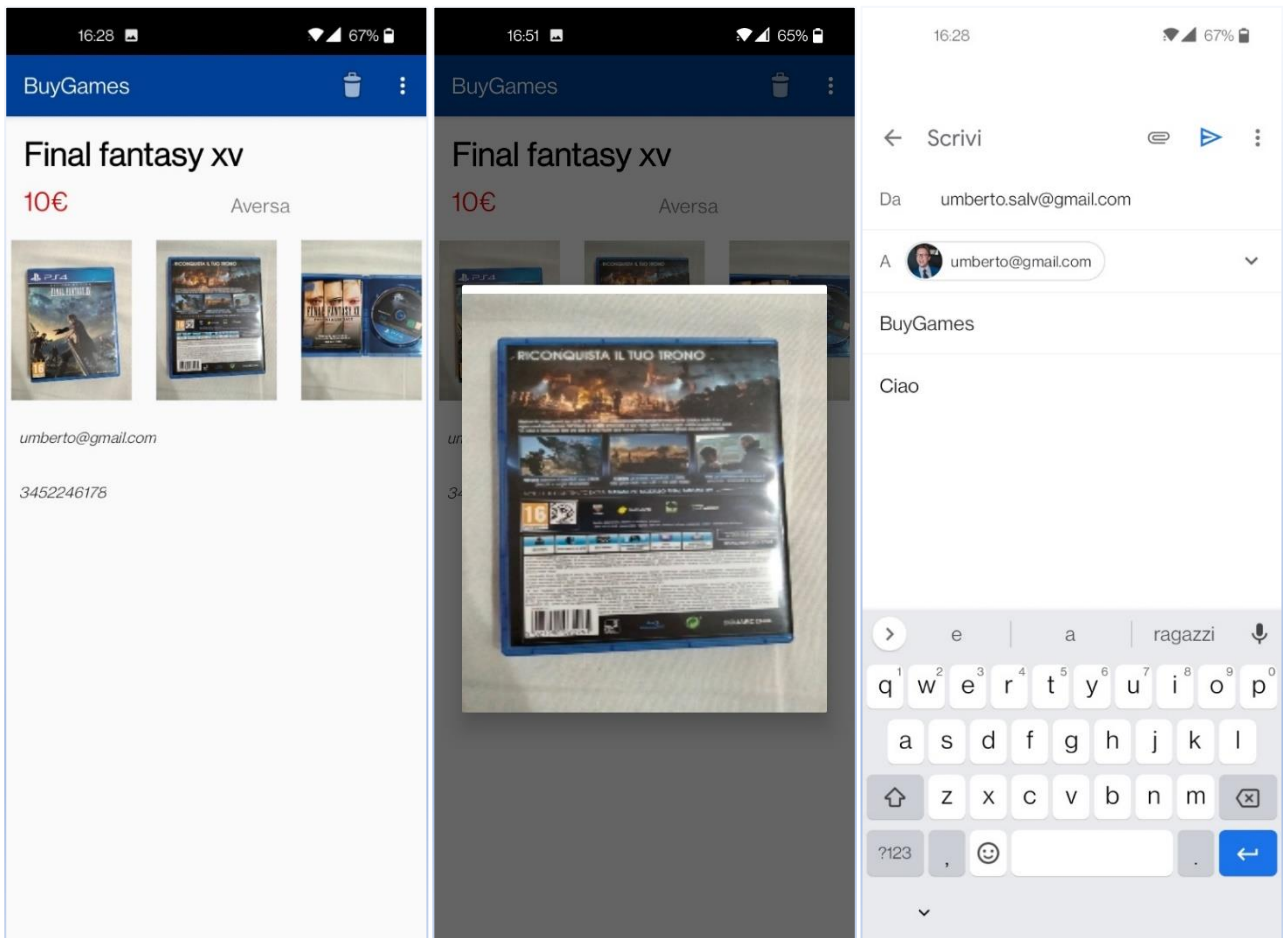
A questo punto sarà possibile cliccare su qualsiasi gioco per vederne le informazioni, oppure, nel caso in cui l'utente sia registrato ed abbia effettuato il *login*, è possibile anche cliccare sul pulsante per inserire un nuovo gioco.

Inoltre, si possono notare:

- Barra di ricerca in alto;
- Menù in alto a destra per effettuare *login* o *logout*;
- Barra di navigazione in basso per accedere all'area personale.

Dettaglio gioco

Una volta selezionato un gioco dalla lista si aprirà la seguente *view* con le varie informazioni:



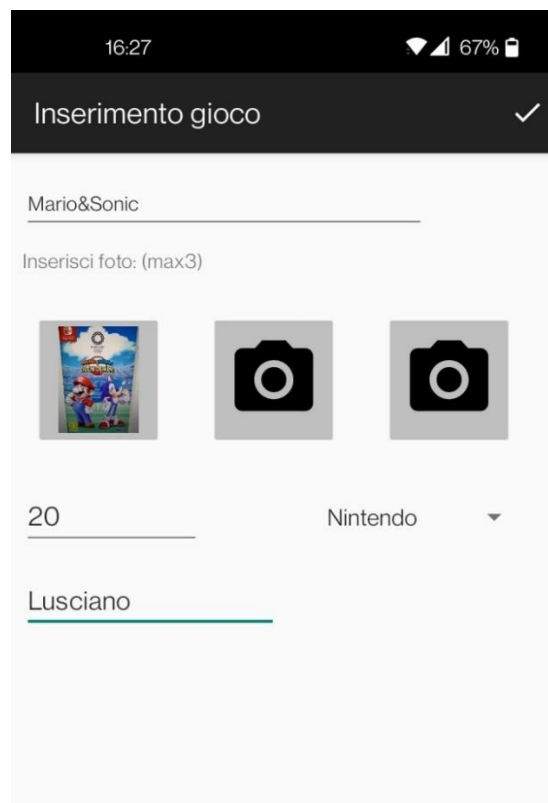
A questo punto è possibile effettuare varie operazioni:

- selezionare una foto per ingrandirla;
- selezionare l'*e-mail* del venditore ed il *software* in automatico precompilerà un'*e-mail* da inviare (come mostrato nell'ultima foto a destra);
- selezionare il numero di telefono per aprire il *dialer* con il numero già inserito;

Inoltre, se l'utente è *loggato* ed il gioco selezionato è stato inserito da egli stesso, in alto a destra potrà scegliere di eliminarlo o selezionare il menu per modificarlo.

Inserimento

Infine, per quanto riguarda l'inserimento, la schermata che l'utente registrato si ritroverà ad utilizzare è la seguente:

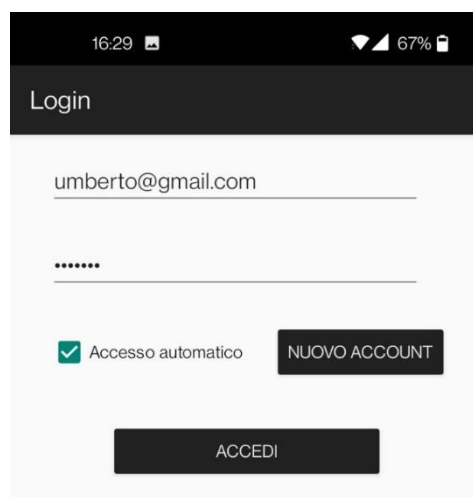


The screenshot shows the 'Inserimento gioco' screen. At the top, the status bar displays the time 16:27, signal strength, and 67% battery. The app header is dark grey with the title 'Inserimento gioco' and a checkmark icon. The form contains the following fields and elements: a text input for the game name with 'Mario&Sonic' entered; a label 'Inserisci foto: (max3)'; three image slots, the first of which contains a game cover; a price input field with '20' entered; a platform dropdown menu currently set to 'Nintendo'; and a text input for the user's name with 'Lusciano' entered.

In cui è possibile inserire: Nome, luogo, prezzo, piattaforma (tramite uno *spinner*) e, opzionalmente, tre foto (nell'esempio è stata inserita solo una foto).

Login

Come accennato spesso è presente la divisione tra Utente ed UtenteRegistrato, quindi è stato necessario creare anche un'interfaccia di *login* nella quale inserire *email* e *password*, scegliere se effettuare l'accesso automatico ogni volta che si apre l'app ed, opzionalmente, si può creare un nuovo account:



The screenshot shows the 'Login' screen. The status bar at the top shows 16:29, signal strength, and 67% battery. The app header is dark grey with the title 'Login'. The form includes: an email input field with 'umberto@gmail.com' entered; a password input field represented by dots; a checkbox labeled 'Accesso automatico' which is checked; a 'NUOVO ACCOUNT' button; and a large 'ACCEDI' button at the bottom.