

SIMONE PEDICO 879307
MANUEL VAGO 887593
SIMONE CORRAIN 887446
CARMELO FERRARO 887447



GAMES UNIVERSE

SOMMARIO

INTRODUZIONE

ARCHITETTURA

FUNZIONALITA'

- Autenticazione
- Libreria giochi
- Ricerca gioco
- Genre
- Profilo utente
- Preferiti

ACTIVITY E FRAGMENT

VIEWMODEL

REPOSITORY

SOURCE

SERVICE

LIBRERIE USATE

CONSIDERAZIONI FINALI E SVILUPPI FUTURI

INTRODUZIONE

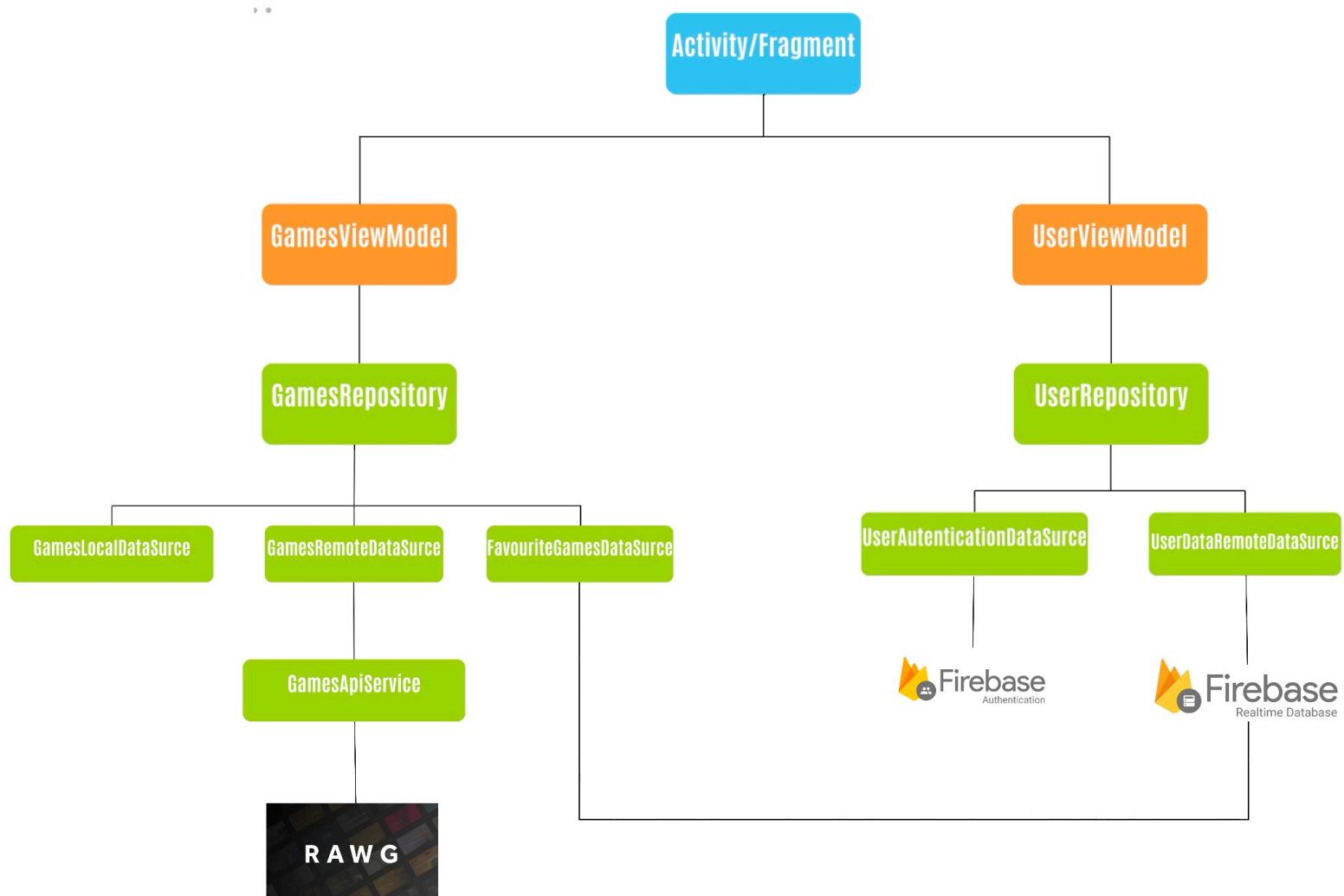
Games Universe è un' app mobile interamente dedicata ai videogiochi. Permette di cercare tra un vasto catalogo, visualizzarne le informazioni e selezionare quelli che l'utente preferisce.

Nasce dalla necessità di un' app dedicata solamente ai videogiochi, in cui trovare informazioni per capire se valga la pena o no acquistarlo. Non esiste ad oggi, infatti, un app che dia informazioni sui videogiochi e bisogna cercarli su internet o vedere video su YouTube.

Al giorno d'oggi l'industria dei videogiochi ha raggiunto vette straordinarie, coinvolgendo milioni di persone. Games Universe è il compagno perfetto per ogni appassionato; questa app è progettata per fornirti le informazioni più recenti, notizie e dettagli su tutti i giochi che ami.

Grazie alla vasta raccolta di dati aggiornati costantemente, terrà informato l'utente su nuovi rilasci e su grandi classici. Esplora un vasto catalogo di giochi, dai classici intramontabili alle ultime uscite.

ARCHITETTURA



L'architettura segue il modello MVVM (Model-View-ViewModel).

Di seguito una breve descrizione del disegno (tutte le componenti sono spiegate nello specifico successivamente).

La parte User gestisce l'autenticazione degli utenti tramite:

- 1) UserAutenticationDataSource: gestisce l'autenticazione tramite FirebaseAuth
- 2) UserDataRemoteDataSource: gestisce i dati degli user appoggiandosi a FirebaseDatabase.

Per quanto riguarda i database i nostri dati sono contenuti nel realtime database, mentre i dati dei videogiochi vengono presi da RAWG.

L'autenticazione è affidata a Firebase authentication.

La parte Games gestisce invece la libreria di videogiochi, divisa in 3 parti:

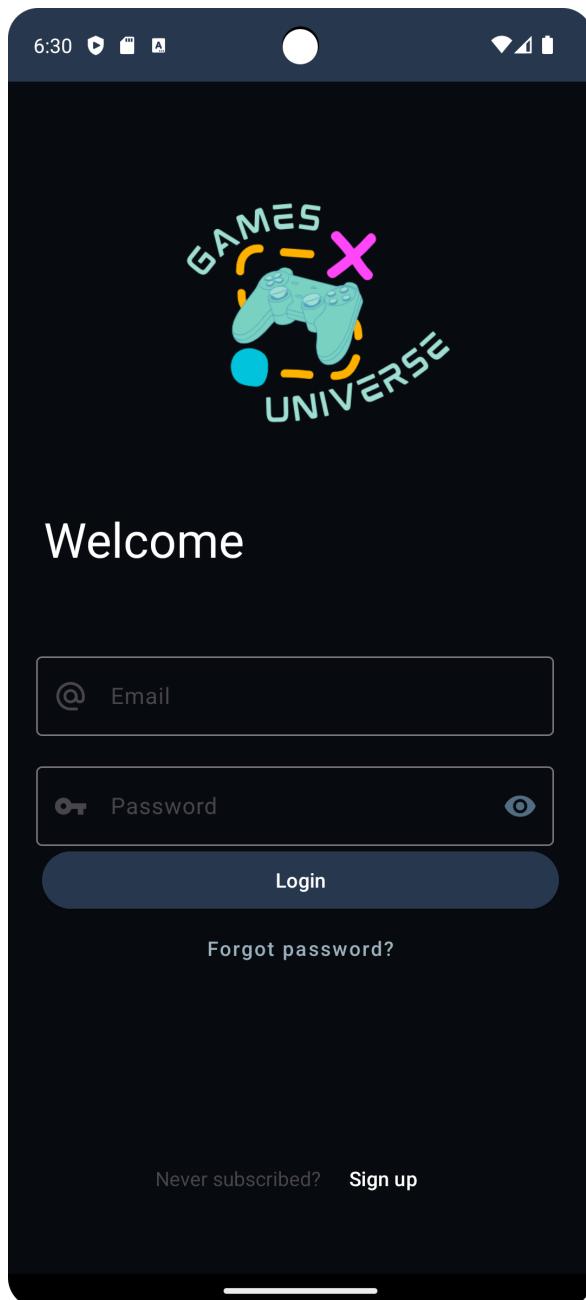
- 1) GamesLocalDataSource gestisce il flusso di dati col database database locale
- 2) FavouriteGamesDataSource: gestisce i videogiochi preferiti recuperandoli dal realtime database
- 3) GamesRemoteDataSource: gestisce effettivamente l'intera libreria dei videogiochi, appoggiandosi a RAWG, un database online che contiene le informazioni su tutti i videogiochi.

FUNZIONALITA' AUTENTICAZIONE

WELCOME

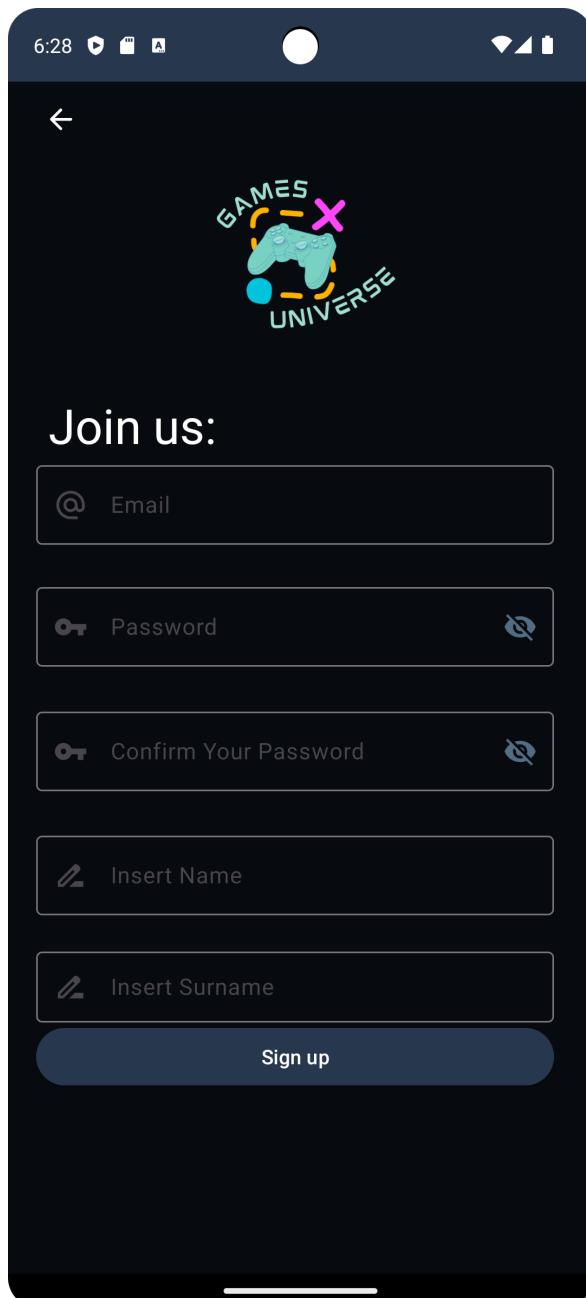
L'utente si autentica tramite una email (che deve essere per forza esistente) e una password (lunga almeno 8 caratteri).

Da questa schermata l'utente già registrato può accedere all'app, inserendo la propria email e la password



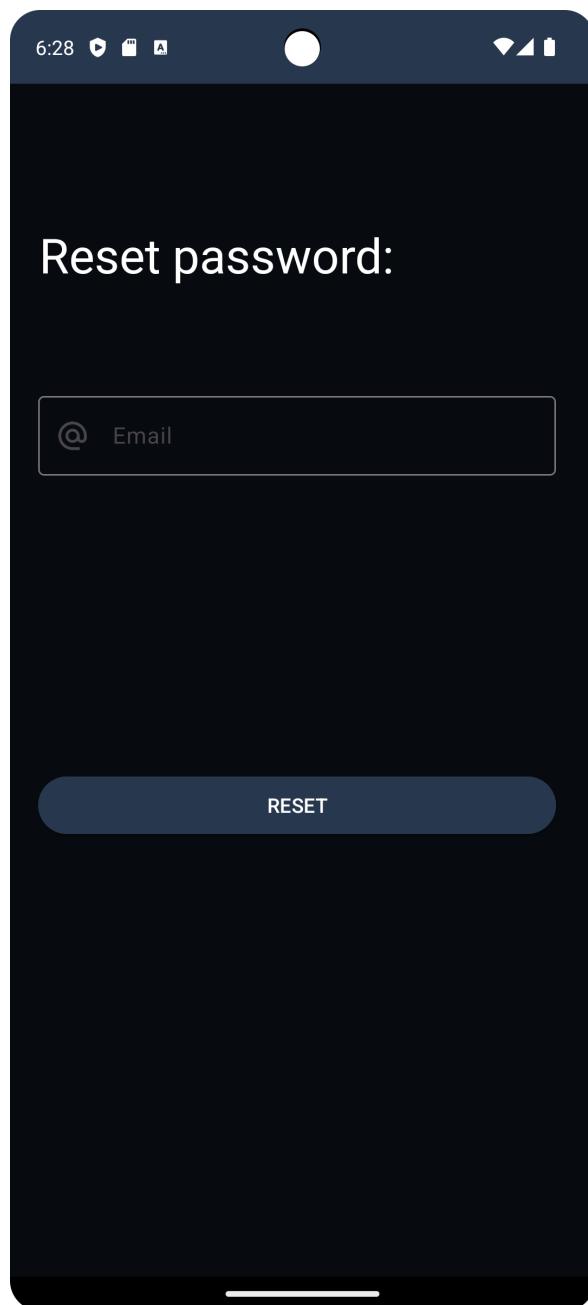
Sign up

Direttamente dalla schermata precedente un nuovo utente può registrarsi, inserendo email, password nome e cognome.



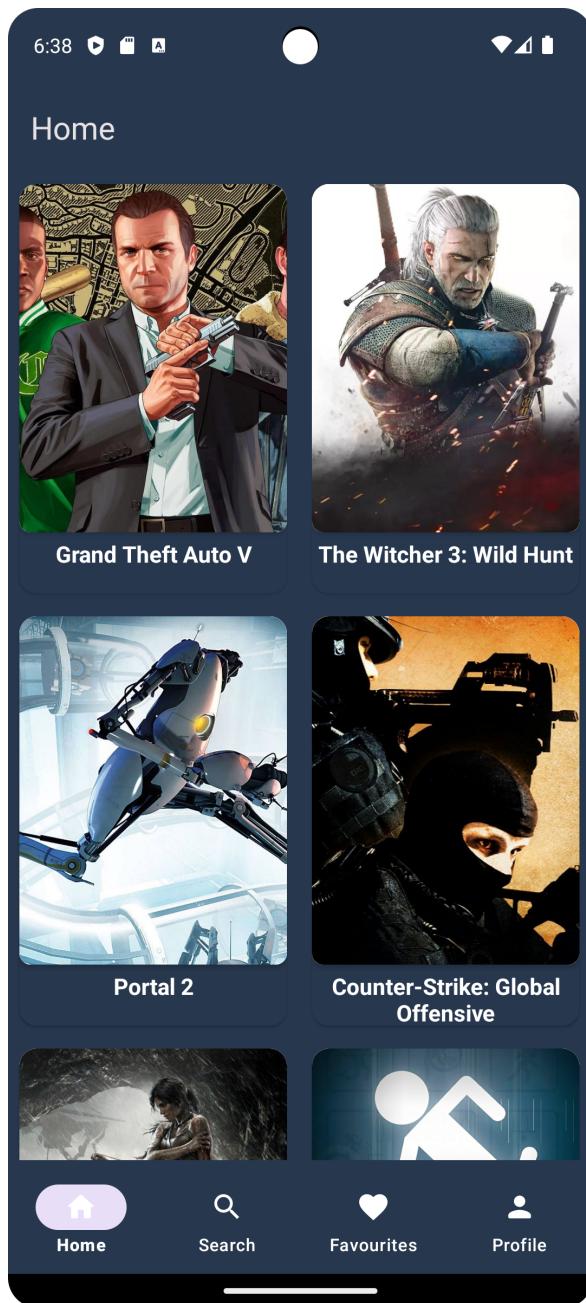
Forgot password

Dalla schermata di login un utente già registrato può accedere alla schermata dedicata al cambio password, tramite l'apposito bottone. Qui potrà cambiare la password con una nuova, il cambiamento viene gestito da firebaseAuthenticator



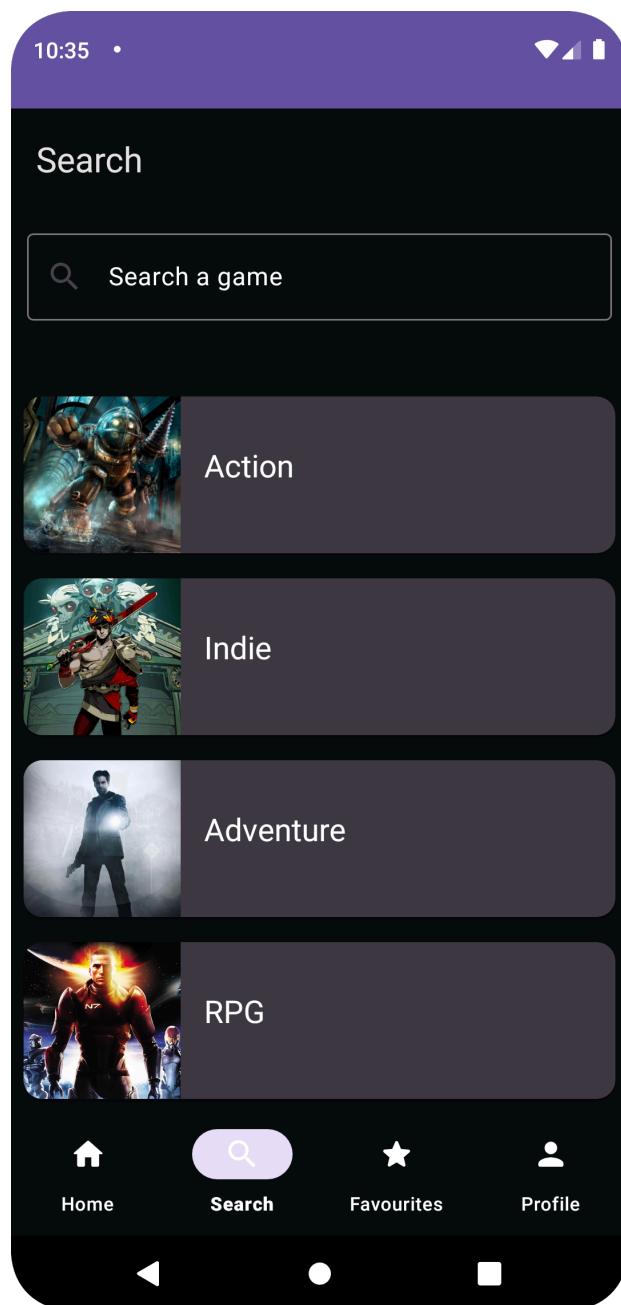
LIBRERIA GIOCHI

Questa è la schermata principale, dove vengono visualizzati i videogiochi.
E' possibile accedere a questa schermata tramite l'apposito bottone nella bottom navigation menu.



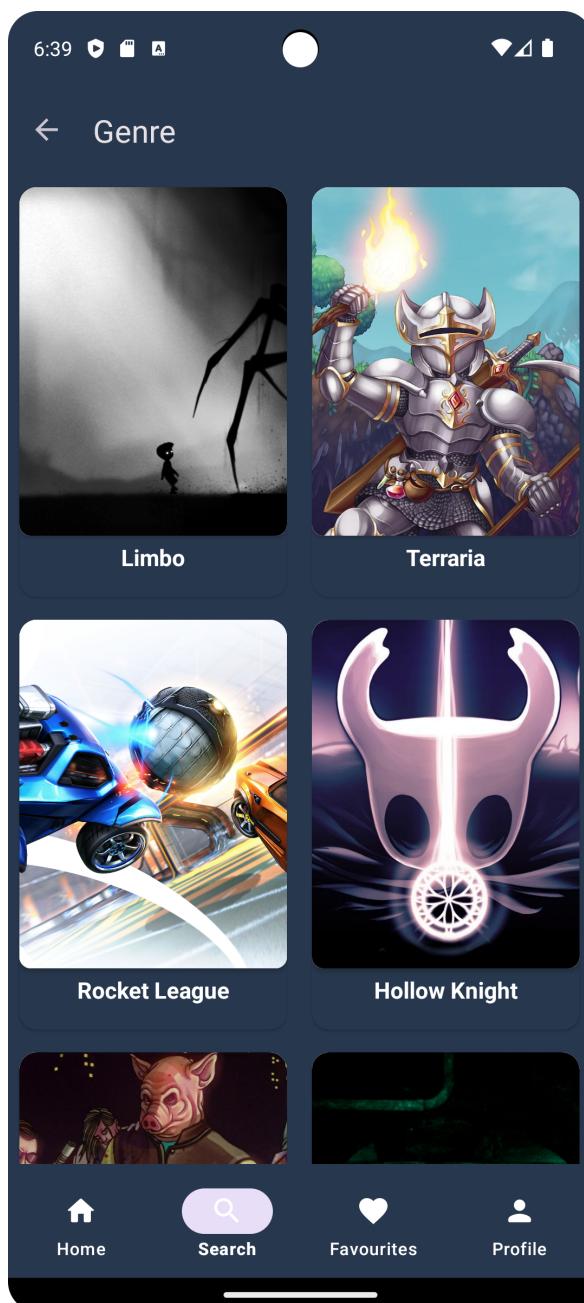
RICERCA GIOCO

Permette di cercare un specifico videogioco inserendone il nome oppure cercare tutti i videogiochi di una determinata categoria tramite gli appositi bottoni.



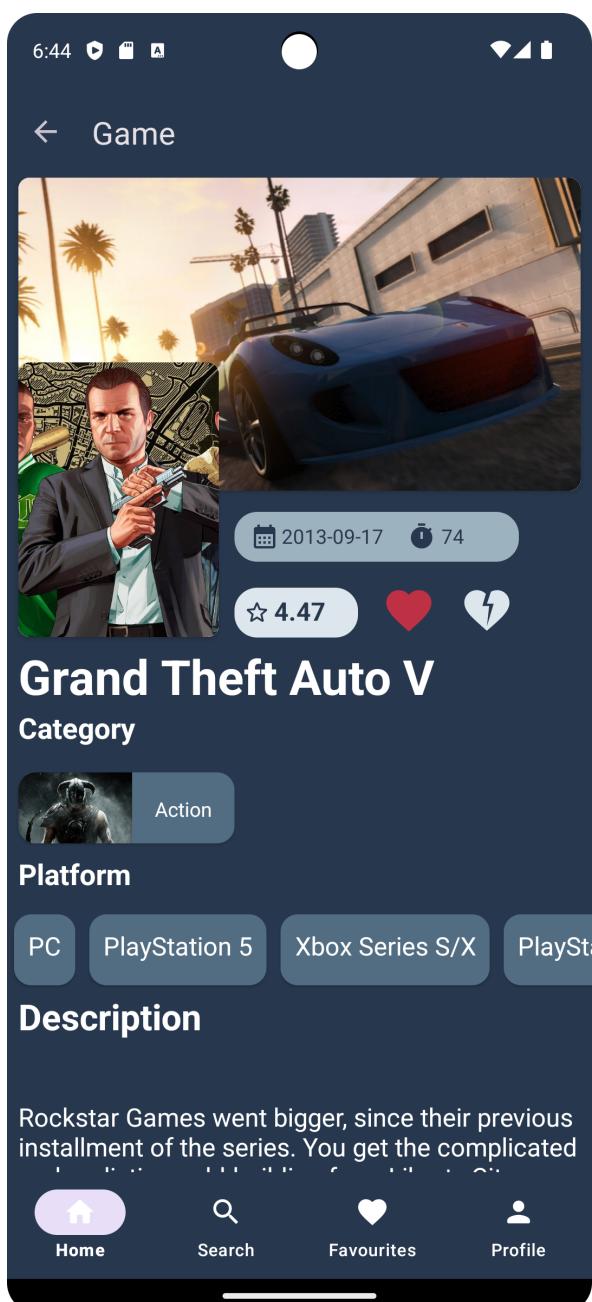
GENRE

Visualizza i videogiochi di un genere, selezionato tramite la schermata di ricerca videogiochi cliccando uno degli appositi pulsanti.



SCHEMATA GIOCO

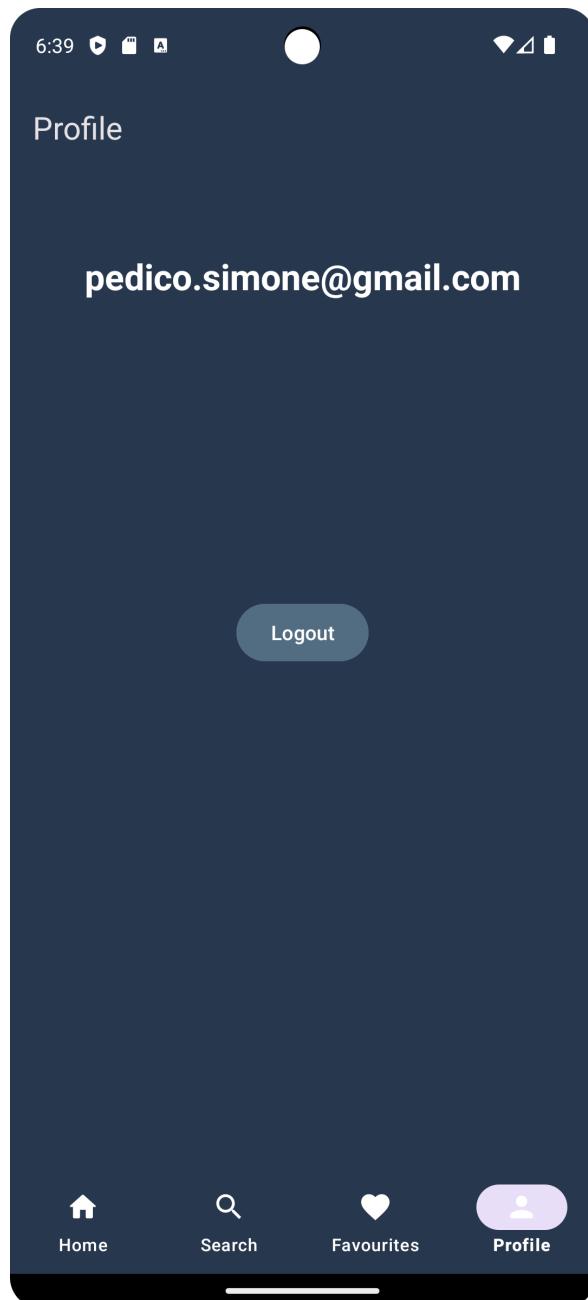
Schermata dedicata al singolo videogioco. Qui vengono visualizzate le informazioni che riguardano il titolo, come la descrizione, la data di uscita, la valutazione media e le ore medie di gioco.



PROFILO UTENTE

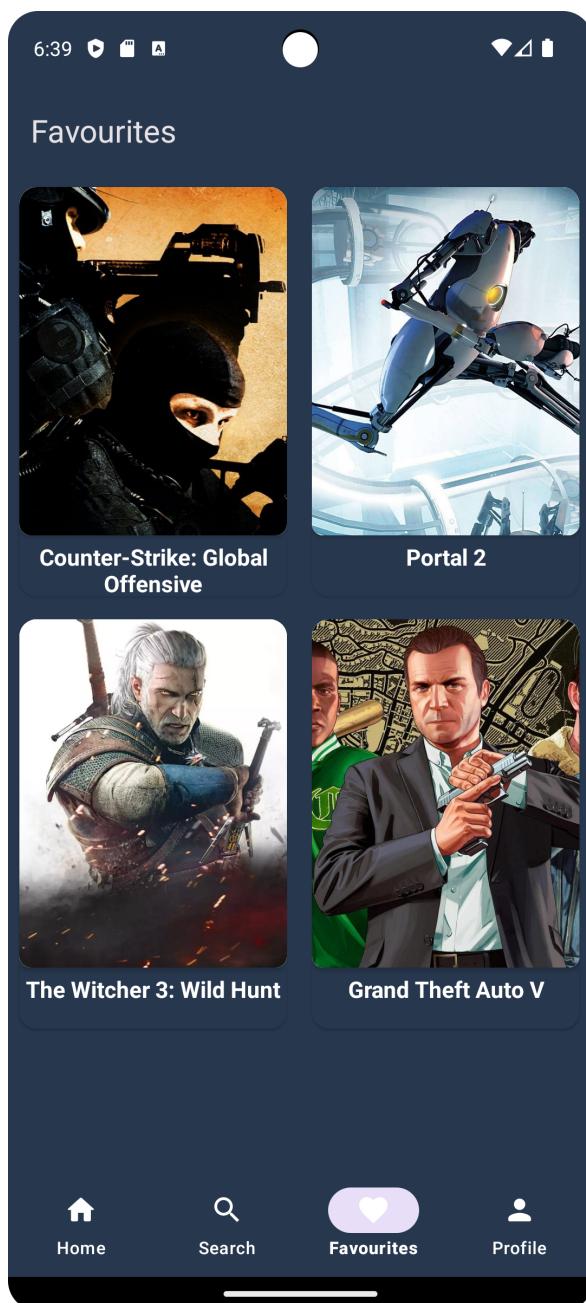
Schermata dedicata al profilo dell'utente, dove viene visualizzata la mail e c'è il tasto per il logout.

Anche a questa schermata si accede con l'apposito bottone nel bottom navigazione menù.



PREFERITI

Permette di visualizzare i videogiochi selezionati dall'utente come preferiti. Vi si accede tramite l'apposito pulsante nel bottom navigation bar.



ACTIVITY E FRAGMENT

WELCOME

LoginFragment

Gestisce la login di un utente già registrato, controllando prima se le mail e la password inserite sono plausibili (email esistente e password lunga almeno 8 caratteri) In caso negativo segnala un errore e non permette l'accesso all'app. Per l'autenticazione si appoggia a UserModel che a sua volta si appoggia a UserAuthenticationDataSource.

SignUpFragment

Contenuto in SignInActivity gestisce la sign up, cioè la registrazione di un nuovo utente nel DataBase. Richiede email, password, nome e cognome.

Inserisce i dati nel realtime database appoggiandosi a UserViewModel che a sua volta si appoggia a UserDataRemoteDataSource

ForgotPasswordFragment

Gestisce la modifica della password; chiede di inserire una nuova password che rispetti i criteri.

MAIN

FavouritesFragment

Contenuto nella MainActivity, permette di visualizzare i videogiochi preferiti dell'utente. Cliccando su di essi visualizza i dettagli del singolo videogioco appoggiandosi a GamesDetailFragment.

GameDetailFragment

Contenuto nella MainActivity, permette di visualizzare le informazioni sul singolo prese dal database online.

GenreFragment

Contenuto nella MainActivity, permette di visualizzare i videogiochi di un dato genere; ogni videogioco può avere più di un genere. Cliccando su un videogioco visualizza i dettagli appoggiandosi a GamesDetailFragment.

HomeFragment

Contenuto nella MainActivity, gestisce la schermata home permettendo di visualizzare i videogiochi tramite il meccanismo di lazy loading. Cliccando su un videogioco visualizza i dettagli appoggiandosi a GamesDetailFragment.

ProfileFragment

Contenuto in MainActivity, gestisce la schermata del profilo del giocatore...

SearchFragment

Contenuto in MainActivity, permette di visualizzare i generi dei videogiochi e cliccando su uno di essi visualizza tutti i videogiochi di quel genere passando alla GenreFragment. Permette anche di cercare un singolo videogioco inserendone il nome.

VIEWMODEL

UserViewModel

Permette la persistenza dei dati degli utenti.

GamesViewModel

Permette la persistenza dei dati dei videogiochi.

MODEL USER

User

Definisce il generico user; ogni user ha nome, cognome, email e password. Contiene anche i metodi necessari per poter ottenere le informazioni dell'user e per poterlo salvare.

GAMES

Game

Definisce alcuni attributi del generico videogioco. I più importanti sono lo slug (tramite cui avviene la ricerca), il nome e l'immagine di copertina.

GameDetail

Definisce il generico videogioco mettendo a disposizione informazioni più rilevanti sul videogioco.

GamesResponse

Classe associata alla risposta di una chiamata API a RAWG.

Genre

Definisce il genere di un videogioco; ogni genere ha associato un nome, un ID e un'immagine.

GenreResponse

Classe associata alla risposta di una chiamata API a RAWG, selezionando i videogiochi di un certo genere.

REPOSITORY USER

UserRepository

Contiene tutti i metodi necessari per poter gestire l'autenticazione e l'inserimento dello user nel database, appoggiandosi a UserAuthenticationRemoteDataSource e UserDataRemoteDataSource.

GAMES

GamesRepository

Contiene tutti i metodi per ottenere le informazioni di un videogioco e per la gestione dei preferiti, appoggiandosi a Games ApiService.

SOURCE USER

UserAutenticationRemoteDataSource

Definisce la logica di autenticazione di un utente tramite Firebase Authentication.

UserDataRemoteDataSource

Definisce la logica di salvataggio di un utente tramite Firebase Realtime Database.

GAMES

FavouriteGameDataSource

Definisce la logica di salvataggio e recupero dei videogiochi preferiti di un utente, tramite Firebase Realtime Database.

GamesLocalDataSource

Definisce la logica di salvataggio e recupero dei videogiochi e dei generi tramite database locale Room.

GamesRemoteDataSource

Definisce la logica di salvataggio e recupero dei videogiochi tramite Games ApiService

SERVICE

Games ApiService

Espone le chiamate API per RAWG.

LIBRERIE UTILIZZATE

Glide

Permette di visualizzare le immagini.

Room

Permette la creazione e l'interazione di database interni dello Storage locale.

Retrofit

Permette la comunicazione mediante HTTP di servizi REST decorando con opportune annotazioni i metodi delle interfacce che rappresentano i servizi da utilizzare.

Firebase

Divisa in due:

- [Firebase Authentication](#): offre un servizio di autenticazione per gli utenti
- [Firebase Realtime Database](#): permette di salvare i dati degli utenti con i relativi giochi preferiti.

Jsoap

Gestisce la formattazione del testo HTML.

CONSIDERAZIONI FINALI E SVILUPPI FUTURI

L'app riguarda un mercato in continua espansione, soprattutto negli ultimi anni. L'idea dietro all'app non è entrare in concorrenza con i grandi produttori di videogiochi, al contrario diventare un mezzo per conoscerne di nuovi dato che non esiste una vera e propria app dedicata a questo.

POSSIBILI AGGIUNTE FUTURE

Una prima possibile aggiunta futura potrebbe essere un forum dedicato a ogni videogioco, in modo che gli utenti possano scambiarsi non solo pareri e idee ma anche consigli, trucchi e spiegazioni.

Una seconda possibile aggiunta potrebbe essere un sistema che assegna un grado per ogni utente, in base a quanto e da quanto usano l'app. In questo modo nei forum sarebbe possibile vedere se a scrivere è un utente nuovo, uno presente da tempo o un veterano in per capirne l'affidabilità.

Un'altra aggiunta potrebbe essere una pagina dedicata al prezzo del videogioco, che confronti le varie offerte online e proponga le più convenienti.

L'idea finale sarebbe quella di creare una community dietro all'app, con utenti che si cambiano pareri e consigli e content creator che scrivono recensioni sui vari giochi. Non più solo un app per conoscere nuovi videogames, ma un vero e proprio social dedicato a questo universo.