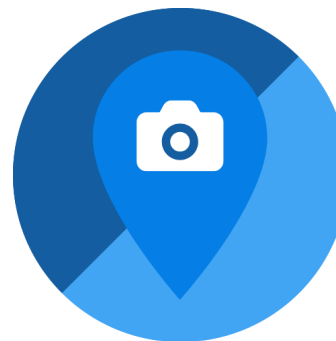


BlackBox Association

Cavaleri Matteo - 875050
Gamba Matteo - 866282
Giura Alberto - 866144
Tombolini Simone - 869564



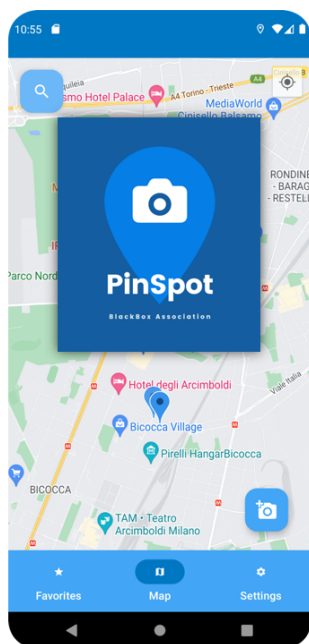
PinSpot

Progetto di Programmazione Dispositivi Mobili – A.A. 2022-23

Università degli Studi di Milano-Bicocca

Introduzione

PinSpot è un'app per dispositivi mobili che permette agli utenti di scoprire, condividere e salvare luoghi interessanti in tutto il mondo.



Ogni utente registrato ha la possibilità di caricare un proprio Pin, con un titolo e un'immagine del luogo, con lo scopo di dare risalto alle bellezze e punti di interesse che il territorio ha da offrire.

Il nome "PinSpot" è l'unione delle due principali funzionalità che l'app mette a disposizione. "Pin" si riferisce ai luoghi presenti sulla mappa, aggiunti da altri utenti, e consultabili da tutti gli utilizzatori dell'app. "Spot" si rifà all'azione di "spottare" un luogo, con l'intento da parte degli utenti registrati di condividerlo.

Funzionalità offerte

L'applicazione fornisce agli utenti la possibilità di registrarsi, creando un profilo con il quale poter sfruttare tutte le funzionalità messe a disposizione. L'utente può comunque scegliere di non registrarsi ed utilizzare l'app in modalità "ospite". In entrambi gli scenari, l'esperienza per entrambi offrirà le funzionalità per:

- Visualizzare i pin sulla mappa in base alla posizione in cui si trova
- Effettuare ricerche per trovare pin in una località differente
- Ottenere informazioni specifiche relative ad un pin, quali foto associate ed il meteo in tempo reale
- Cambiare alcune impostazioni di visualizzazione

Agli utenti registrati è possibile inoltre:

- Aggiungere pin sulla mappa, corredati con foto del luogo e titolo a scelta dell'utente
- Creare un elenco di pin preferiti da poter visualizzare in qualsiasi momento
- Reimpostare la password e disconnettersi

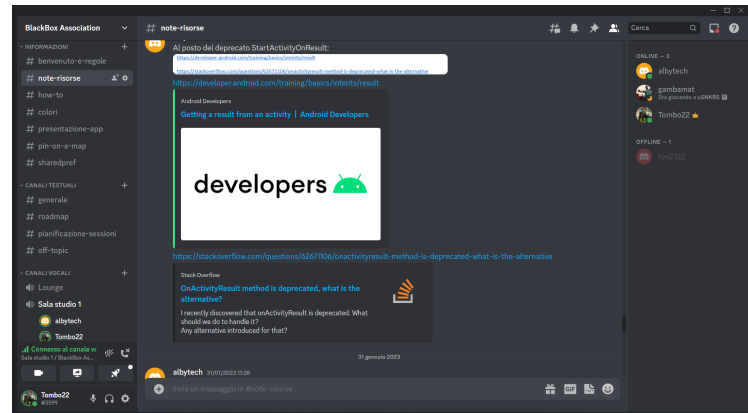
Al primo avvio l'applicazione richiede all'utente di effettuare il login con un account (volendo anche attraverso i servizi Google) oppure continuare come ospite ed accedere quindi in modalità limitata alle funzionalità dell'app.

Una volta raggiunta la schermata della mappa, l'utente potrà "*spottare*" un nuovo luogo, ovvero aggiungere un pin alla mappa, nella posizione in cui si trova l'utente in quel determinato momento, al quale sarà allegata una foto rappresentativa del luogo scoperto; infine, l'utente sceglierà per esso un titolo e lo condividerà con tutti gli altri utilizzatori dell'app.

I pin aggiunti dagli altri utenti sono caricati dinamicamente sulla base dell'interazione con la mappa e attraverso i marker posizionati su di essa sarà possibile accedere alla schermata di dettaglio dei pin per consultare il titolo, la foto e le informazioni meteo presenti in quel luogo, nell'istante della visualizzazione. Inoltre, se di suo gradimento, l'utente potrà aggiungerlo dalla stessa scheda alla lista dei preferiti, per una visualizzazione più rapida in futuro.

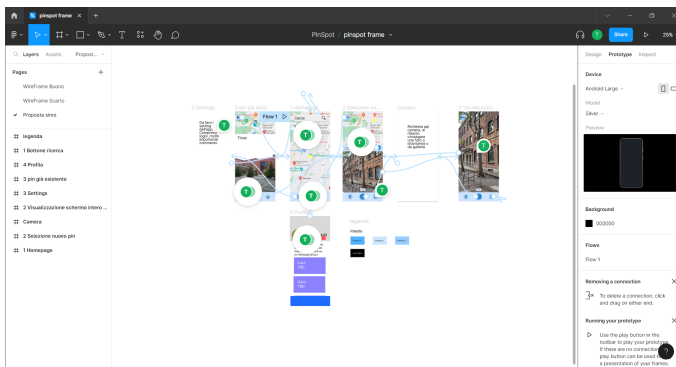
Gestione del progetto

Le riunioni avvenivano periodicamente o in presenza oppure utilizzando l'app Discord per fare il punto della situazione, discutere le azioni future da intraprendere e mostrare il proprio lavoro svolto, condividendolo con il resto del gruppo.



GIT e GitHub sono stati utilizzati per il controllo versione del codice in modo da poter rendere le azioni di cooperazione più facili e veloci tra i componenti del gruppo.

Inizialmente lo sviluppo si è concentrato sulla realizzazione di singole features, che poi pezzo dopo pezzo sono confluite nel progetto finale, tutto ciò per agevolare la comprensione dei vari argomenti visti a lezione.



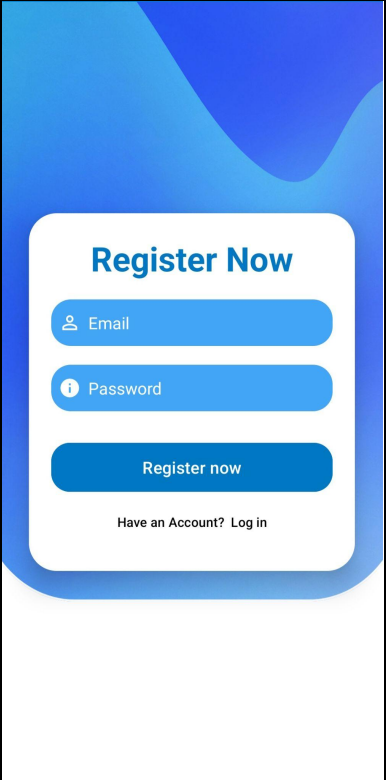
Figma è stato utilizzato in principio solo per la creazione dei wireframe e delle bozze relativi alla UI, per poter comunicare e definire fin da subito le funzionalità e il flusso di navigazione che l'app avrebbe dovuto avere una volta terminata.

Anche se i layout e la grafica sono stati perfezionati successivamente solo su Android Studio, l'utilizzo di Figma è servito come spunto per le idee che poi sarebbero diventate l'app definitiva.

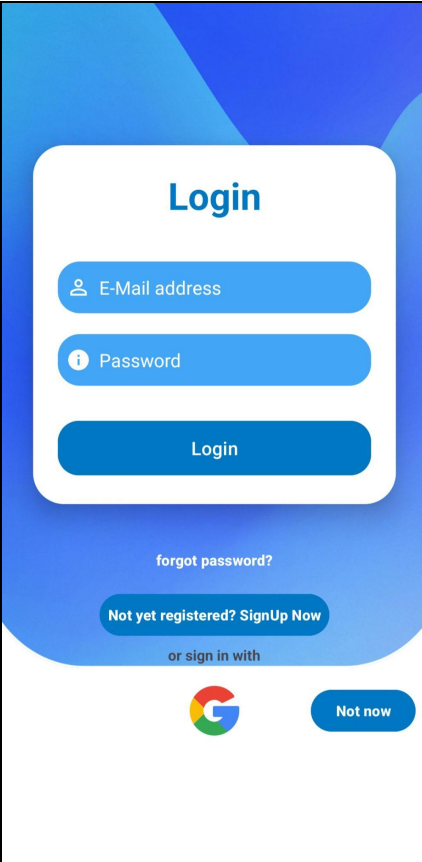
Layout

SignupFragment

Questo fragment consente agli utenti sprovvisti di account di registrarsi per fruire dell'esperienza completa dell'app. I dati richiesti per la registrazione sono l'email personale ed una password.



The Register Now screen features a white card on a blue gradient background. The card has a title 'Register Now' in bold blue text. Below the title are two input fields: 'Email' with a person icon and 'Password' with an information icon. A blue 'Register now' button is positioned below the fields. At the bottom of the card, there is a link 'Have an Account? Log in'.



The Login screen features a white card on a blue gradient background. The card has a title 'Login' in bold blue text. Below the title are two input fields: 'E-Mail address' with a person icon and 'Password' with an information icon. A blue 'Login' button is positioned below the fields. Below the card, there is a link 'forgot password?' and a button 'Not yet registered? SignUp Now'. At the bottom, there is a link 'or sign in with' followed by the Google logo and a 'Not now' button.

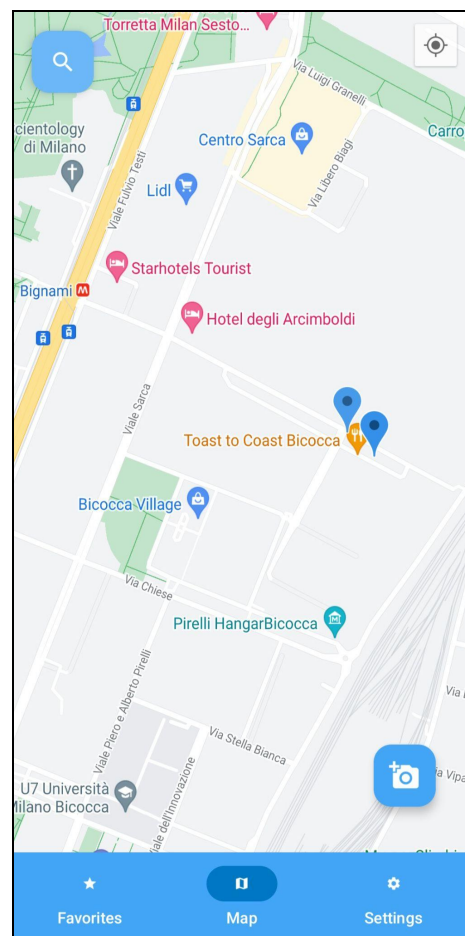
LoginFragment

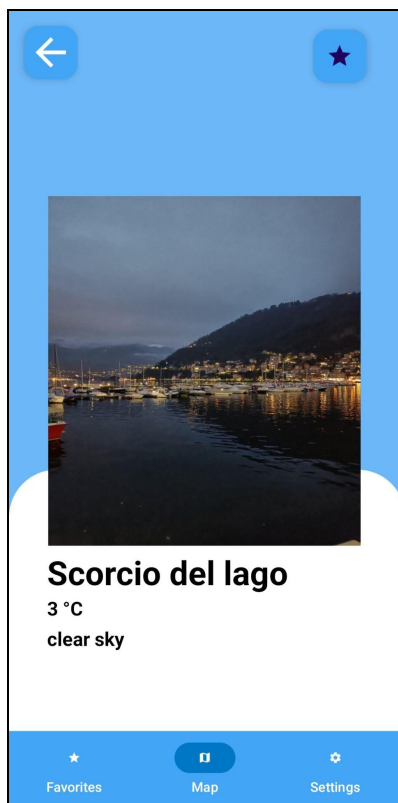
Una volta che l'utente avrà creato il suo account, sarà in grado di accedere a PinSpot inserendo le proprie credenziali. Nel caso in cui avesse dimenticato la password, sarà possibile reimpostarla autonomamente cliccando l'opzione dedicata e fornendo l'indirizzo email utilizzato per la registrazione. È possibile effettuare l'accesso e fruire delle funzionalità complete anche tramite account Google eventualmente registrati sul proprio dispositivo.

All'utente è lasciata altresì la possibilità di saltare la fase di login e registrazione dell'app. Questa scelta comporterà l'impossibilità per l'utente di aggiungere nuovi luoghi alla mappa e di creare la propria lista di pin preferiti.

MapFragment

Rappresenta il fragment di principale rilevanza per l'esperienza dell'utente, costituito dalla mappa fornita da Google Maps, sulla quale vengono caricati i pin inseriti dagli altri utenti. In questa schermata è anche possibile spostare la vista della mappa su un'altra posizione, servendosi della barra di ricerca con completamento automatico, ed individuare così gli eventuali pin presenti nei luoghi desiderati. I pin vengono inoltre recuperati dinamicamente e resi disponibili all'utente man mano che quest'ultimo sposta la visuale della mappa.



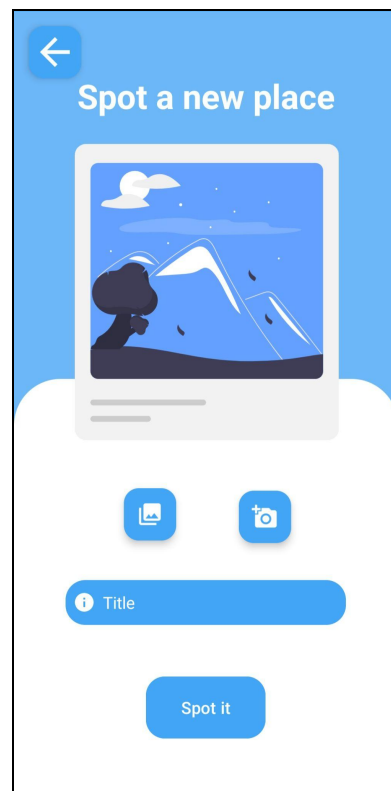


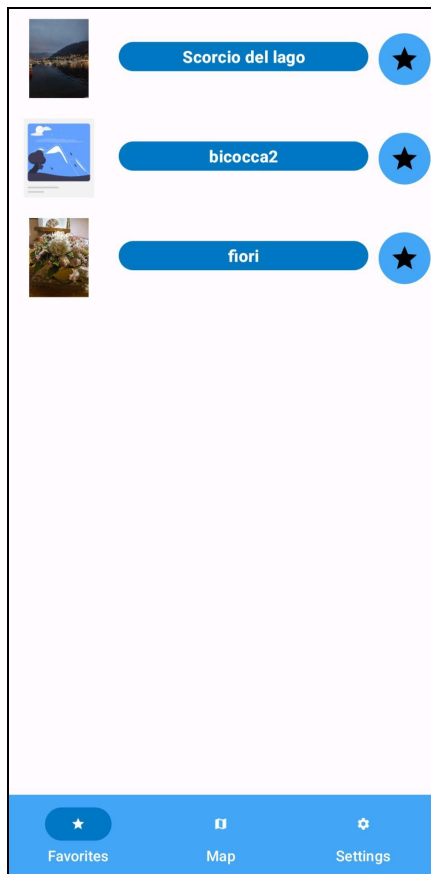
PinInfoFragment

Si tratta della scheda di dettaglio relativa ad uno specifico pin, aperta cliccando il corrispondente marker presente sulla mappa o scegliendo un pin dalla lista preferiti. In essa sono contenuti il titolo assegnato al pin, la foto rappresentativa dello stesso e alcune informazioni di interesse per il luogo, ovvero la temperatura ed il meteo forniti in tempo reale.

InsertPinActivity

L'inserimento di un nuovo pin è una delle funzionalità chiave dell'app: quando l'utente si troverà in una location considerata di interesse fotografico, potrà attraverso questa schermata scattare una foto al momento o selezionarne una dalla galleria da associare al nuovo luogo. Infine, una volta scelto il titolo desiderato, potrà condividere la location, mettendola a disposizione degli altri utenti.





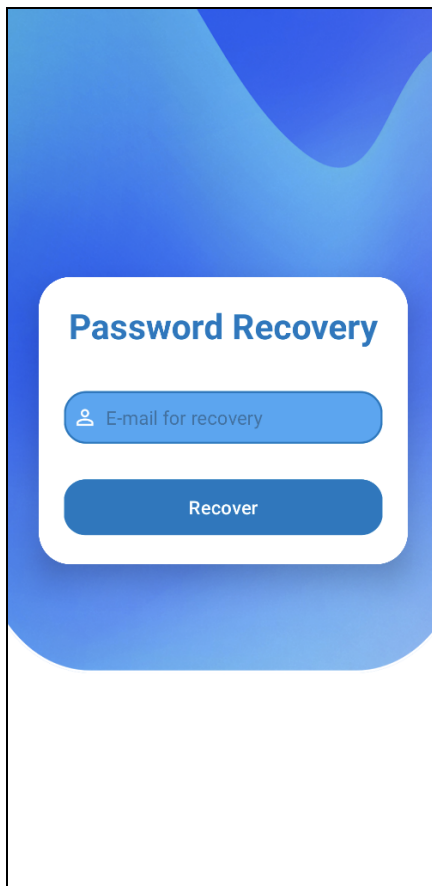
FavPlacesFragment

Questo fragment contiene una RecyclerView utilizzata per mostrare i pin preferiti dell'utente, visualizzati a schermo con titolo e relativa foto del luogo. I contenuti di tale lista sono salvati in locale attraverso Room. I preferiti che vengono aggiunti consentono di raggiungere rapidamente la scheda di dettaglio del singolo pin.

SettingsFragment

Da questa schermata l'utente può visualizzare e modificare alcune impostazioni riguardanti la sua esperienza con l'app, relative alla temperatura nella scheda di dettaglio del pin, al proprio account ed al parametro per l'area di caricamento dinamico dei pin.



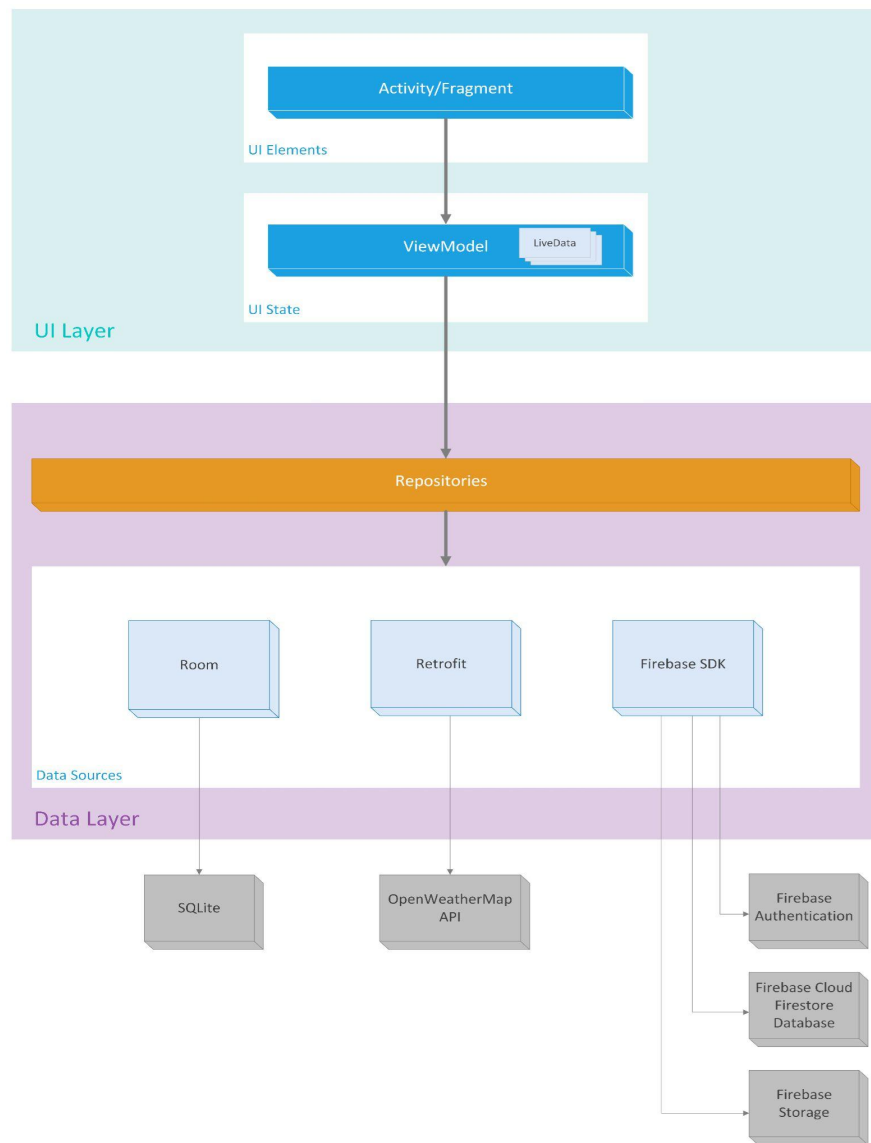


ForgotPasswordFragment

Nel caso in cui l'utente non ricordasse la propria password, potrà semplicemente inserire la mail utilizzata per la registrazione e ricevere su di essa un link per provvedere a reimpostarla.

Architettura dell'app

L'architettura ricalca le linee guida della documentazione Android, implementando il pattern "Model-View-ViewModel". Il disegno sottostante rappresenta in sintesi l'intera architettura dell'app, che andremo poi a descrivere più nel dettaglio solo nelle sue componenti fondamentali.



Qui di seguito l'elenco di tutte le classi inserite nel progetto divise per package:

Route			Nome Classe
Adapter			PinRecyclerViewAdapter
Data	DataBase		PinDao
			PinRoomDatabase
	Repository	Pin	IPinRepository
			PinRepository
		User	IUserRepository
			UserRepository
		Weather	IWeatherRepository
			WeatherRepository
		Service	WeatherApiService
	Source	Pin	BaseFavoritePinLocalDataSource
			BasePinRemoteDataSource
			FavoritePinLocalDataSource
			PinCallback
			PinRemoteDataSource
		User	BaseUserAuthRemoteDataSource
			UserCallback
			UserAuthRemoteDataSource
		Weather	BaseWeatherRemoteDataSource
			WeatherCallback
			WeatherRemoteDataSource
Model			WeatherApiResponse
			Pin

			Result
			User
Ui	Main		FavPlacesFragment
			InsertPinActivity
			MainActivity
			MapFragment
			MapViewModel
			PinInfoFragment
			PinViewModel
			PinViewModelFactory
			SettingsFragment
			WeatherViewModel
			WeatherViewModelFactory
	Welcome		LoginActivity
			LoginFragment
			SignUpFragment
			ForgotPasswordFragment
			UserViewModel
			UserViewModelFactory
Util			Constants
			GlideModuleUtil
			ServiceLocator

WeatherApiResponse: classe che riproduce la struttura della risposta JSON fornita dalla chiamata alla OpenWeatherMap.org API, opportunamente annotata per consentire il parsing della risposta attraverso Gson.

WeatherCallback: interfaccia utilizzata dalla WeatherRepository per gestire attraverso delle callback i casi di successo/fallimento provenienti dalla WeatherRemoteDataSource.

WeatherRepository: classe che espone a WeatherViewModel i metodi per recuperare le informazioni relative alla temperatura e meteo di un dato Pin attraverso le sorgenti di dati, ovvero in questo caso WeatherRemoteDataSource.

WeatherViewModel: classe dedicata alla separation of concerns tra UI e Repository. È a disposizione della classe PinInfoFragment, che la utilizza per ottenere informazioni relative allo specifico pin aperto.

WeatherViewModelFactory: si occupa di creare istanze di WeatherViewModel, a partire dalla stessa WeatherRepository.

WeatherApiService: fornisce l'interfaccia per eseguire un'interrogazione alla OpenWeatherMap.org API.

Result: classe di supporto istanziata, nelle sue implementazioni concrete, a seguito di un risultato di successo o fallimento ed utilizzata come tipo di dato per i MutableLiveData.

Pin: classe che definisce la struttura dell'oggetto Pin, la cui interpretazione visiva sulla mappa corrisponde ad un marker. Un pin è caratterizzato dall'avere un titolo, le sue coordinate geografiche (latitudine e longitudine), un riferimento alla foto associata a

tale luogo ed un geohash, ovvero un parametro di supporto generato a partire dalle coordinate geografiche ed utilizzato per eseguire query sul database remoto.

PinDao: interfaccia che associa query SQL rivolte al database locale (implementato con Room) a metodi, attraverso annotazioni.

PinRoomDatabase: classe astratta che semplifica il lavoro del database.

PinRecyclerViewAdapter: adapter realizzato per interagire con gli elementi della RecyclerView.

Librerie e Servizi

Le librerie utilizzate per poter sviluppare l'app sono le seguenti:

- **Firebase Auth:** servizio impiegato per consentire la registrazione degli utenti nell'app attraverso un'opzione di login con email e password oppure tramite accesso diretto con Account Google.
- **Firebase Cloud Firestore:** database utilizzato per salvare i dati relativi ai nuovi luoghi spottati e recuperare i pin da mostrare sulla mappa. Dopo alcune prove con Firebase Realtime Database, siamo giunti alla conclusione di preferire Firestore a tale soluzione, motivati dalla necessità di dover recuperare pin sulla mappa eseguendo un particolare tipo di query, le GeoQueries, che potessero determinare se un pin si trovasse o meno all'interno di un quadrante geografico, individuato a partire da una data posizione ed un raggio, inteso come distanza massima da tale posizione.

- **Firebase Storage:** servizio di archiviazione cloud fornito da Firebase, nel quale vengono inserite le foto relative ai singoli pin e che sono ad essi associate attraverso l'attributo "link" degli stessi.
- **Glide:** libreria implementata per recuperare le immagini associate ai vari pin da Firebase Storage.
- **Retrofit:** client REST per Java e Android. Attraverso tale libreria, congiuntamente con Gson, si interagisce con la OpenWeatherMap.org API e si gestisce il processo di creazione di chiamate e ricezione di risposte HTTP.
- **Gson:** libreria per la serializzazione/deserializzazione tra oggetti Java e JSON.
- **Google Play Services:** dipendenze aggiunte per fornire l'autenticazione tramite Account Google.
- **Google Maps SDK:** utilizzata per fornire agli utenti la mappa sulla quale visualizzare i pin inseriti da altri utenti e dare la possibilità di posizionare i propri.
- **Google Places SDK:** impiegata per fornire la barra di ricerca, comprensiva della funzionalità di auto-completamento per suggerimenti, così da permettere lo spostamento sulla mappa ad un'altra località nella quale sia possibile verificare l'esistenza di altri pin.

L'API utilizzata

La OpenWeatherMap API è stata utilizzata per recuperare in tempo reale la temperatura ed il meteo corrente di un determinato pin, nel momento in cui l'utente fa clic su di esso sulla mappa e ne consulta la relativa scheda di dettaglio.

Sviluppi futuri

Black Box Association si pone come obiettivo di continuare ad aggiornare e migliorare PinSpot nei seguenti aspetti:

- Notifiche eventi futuri: notifiche push per avvisare l'utente quando un Pin viene aggiunto nel raggio di 5 km dalla posizione corrente.
- Notifiche custom: attraverso algoritmi di machine learning inviare una notifica speciale quando viene postato un Pin che molto probabilmente è di forte interesse all'utente.
- NetWorking: vuole essere fornita la possibilità di poter interagire con altri utenti potendo segnalare tramite like un post particolarmente bello.
- NetWorking(2): vuole essere fornita la possibilità di comunicazione tra gli utenti dell'app mediante una chat personale e/o di gruppo.
- Pubblicazione sul Play Store: dopo aver rifinito alcuni dettagli e magari realizzato alcuni degli altri sviluppi futuri, ci piacerebbe rilasciare l'applicazione sul Play Store. Fino a quel momento, sarà possibile installarla tramite APK o compilarla autonomamente a partire dal codice sorgente, disponibile pubblicamente su GitHub.